uc3m | Universidad Carlos III de Madrid

Grado Universitario en Ingeniería
Electrónica Industrial y Automática

Curso 2016-2017

Trabajo Fin de Grado

"Desarrollo de herramienta para etiquetado denso de escenas"

Inés M. Carpio Coll

Tutor: Jorge Beltrán de la Cita

Leganés, octubre 2017

AGRADECIMIENTOS

A mi familia, por todo el cariño
y el apoyo incondicional con el que he crecido,
que me ha ayudado llegar a ser quien soy hoy.

A Sergio, por llevar cuatro años
siendo el compañero perfecto en
tantos aspectos de mi vida.

A los profesores que me han formado
hasta el día de hoy, en especial a mi tutor Jorge,
por su simpatía y su paciencia durante estos meses.

RESUMEN

Este Trabajo de Fin de Grado (TFG) desarrolla una aplicación de etiquetado denso de imágenes. Esta herramienta permite un proceso de anotación masiva de bases de datos de una manera sencilla e intuitiva, con el objetivo de minimizar los tiempos y costes asociados a este tipo de tareas.

La aplicación consta de una interfaz gráfica de usuario que permite el etiquetado de imágenes de forma sencilla. Con este objetivo, el usuario debe señalar puntos sobre los contornos de los diferentes elementos de la imagen que desee etiquetar para ir formando polígonos que contengan los pixeles correspondientes con cada una de las categorías a anotar. El conjunto de etiquetas utilizadas para realizar las anotaciones es completamente configurable, lo que amplía las áreas de aplicación de la herramienta desarrollada, no limitándola a las bases de datos para conducción autónoma.

Por último, y con el fin de facilitar la integración de esta herramienta en los actuales flujos de trabajo de los investigadores, las anotaciones pueden exportarse en múltiples formatos, incluyendo aquellos utilizados en las bases de datos de conducción autónoma más populares.

Los resultados obtenidos son satisfactorios, habiéndose logrado completar todos los objetivos planteados. Además, se ha conseguido implementar una herramienta intuitiva y sencilla de utilizar, pensada para facilitar y agilizar el proceso de etiquetado masivo de imágenes.

Palabras clave: Herramienta de etiquetado, Interfaz Gráfica de Usuario, Visión por computador, Bases de datos, Vehículos autónomos.

ABSTRACT

This Bachelor Thesis develops an image dense labelling tool. This software allows a mass database annotation process in a simple and intuitive way, with the aim of minimizing the time and costs associated with this type of tasks.

The application consists of a graphical user interface that allows the labelling of images in a simple way. With this objective, the user must point out points on the contours of the different elements of the image that he wants to label, to form polygons containing the corresponding pixels for each of the categories to be annotated. The set of labels used to make annotations is fully configurable, which extends the application areas of the tool developed, not limited to databases for autonomous driving.

Finally, in order to facilitate the integration of this tool into the current workflows of researchers, annotations can be exported in multiple formats, including those used by the most popular autonomous driving databases.

The results obtained are satisfactory, having been able to complete all the objectives set. An intuitive and simple-to-use tool has been implemented, designed to facilitate and accelerate the process of mass labelling of images.

Keywords: Labelling tool, Graphical User Interface, Computer vision, Databases, Autonomous vehicles.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

1. INTRODUCCIÓN

La tecnología es la ciencia aplicada a la resolución de problemas o a la mejora de la situación actual. El creciente interés por el desarrollo de los vehículos de conducción autónoma podría suponer la solución para varios problemas existentes, como la contaminación medioambiental o el alto número de muertes por accidentes de tráfico en todo el mundo. Además, estos vehículos prometen ser una auténtica revolución en el mundo automovilístico, por lo que las grandes compañías que operan en este mercado, y algunas tecnológicas, están actualmente en vías de desarrollo de la tecnología necesaria para este tipo de conducción, con el objetivo de que en el año 2020 existan vehículos comerciales completamente autónomos en las calles de todo el mundo.

Para conseguir este hecho, es necesario un gran sistema de control y procesamiento que informe al vehículo de su entorno y tome las decisiones apropiadas para navegar. Una de las áreas más importantes en este sistema es la visión artificial o visión por computador, que engloba los procesamientos de las imágenes capturadas para modelar el entorno del vehículo, detectando y clasificando los diferentes elementos de la escena. Para realizar este proceso se necesitan algoritmos de inteligencia artificial que sean capaces de funcionar en los diferentes entornos de conducción. Por ello es imprescindible contar con bases de datos anotadas extensas y variadas para poder entrenar correctamente dichos algoritmos, de tal manera que generalicen su aprendizaje y funcionen de manera precisa en todos los entornos posibles.

Debido a esa demanda de bases de datos anotadas, y a la falta de herramientas estandarizadas para generarlas, se ha decidido crear una herramienta que facilite el etiquetado de nuevas bases de datos de imágenes, enfocándose sobre todo en las necesidades de las plataformas de investigación que actualmente se están llevando a cabo en el Laboratorio de Sistemas Inteligentes (LSI) de la Universidad Carlos III.

1.1. OBJETIVOS

El objetivo de este trabajo es desarrollar una herramienta que permita el etiquetado masivo de bases de datos de imágenes, de tal forma que las anotaciones resultantes sean utilizadas para entrenar los vehículos autónomos que se están desarrollando en el LSI.

El principal objetivo de la herramienta es desarrollar una interfaz gráfica que permita a los investigadores del laboratorio etiquetar todas las imágenes tomadas con los vehículos. Esta tarea puede llegar a ser larga y tediosa, por lo que en la implementación de esta herramienta se busca conseguir agilizar el proceso al usuario, hacerlo sencillo, intuitivo y que simplifique el etiquetado de las imágenes.

La herramienta debe cumplir las siguientes características:

- Permitir diferentes formatos de entradas: el usuario debe poder abrir un vídeo o una secuencia de imágenes, además de poder navegar a través de ellas (o de los frames del vídeo) tanto hacia delante como hacia atrás.
- Disponer de una sección de categorías denominadas Etiquetas que sean totalmente configurables y que cuenten con un nombre, un número ID que representa el nivel de gris, un color y con las opciones binarias (activado o desactivado) de las propiedades de ser instanciable y de asignar un *BoundingBox*. Estas etiquetas deben poder crearse desde cero y después poder editar sus atributos o ser eliminadas. También debe existir la posibilidad de cargar las etiquetas desde un archivo ya existente.
- Permitir un etiquetado denso sencillo de las diferentes categorías en cada imagen de entrada. Para ello, el usuario debe poder seleccionar puntos en la imagen cargada, que representen los vectores de un polígono. Este polígono está asociado a una etiqueta, por lo que es dibujado en la imagen con el color asignado a la misma.
- Ofrecer diferentes formatos de salida para las anotaciones realizadas. Entre ellos, deben encontrarse los formatos más habituales ya utilizados en otras bases de datos con fines similares: una imagen donde estén los polígonos dibujados a

color, otra con los niveles de gris y otra con los polígonos asociados a las etiquetas que tengan activada la opción de *Instanciable*. También se debe poder exportar la posición de los polígonos en un documento de XML y, de igual manera, poder exportar las etiquetas que existen en otro XML, que guarde los atributos de las mismas.

1.2. PLANIFICACIÓN

La planificación temporal del proyecto se ha dividido en distintas secciones, con el fin de organizar el tiempo dedicado a este proyecto. Para la estimación de tiempo se tuvo en cuenta que durante los meses de abril y mayo existirían días en los que no se pudiera adelantar el proyecto por temas académicos, así como pequeños parones por temas personales durante los siguientes meses.

Para una mejor visualización del proceso planificado en el tiempo, se hace uso de la herramienta gráfica del Diagrama de Gantt [1] (Figura 2), la cual facilita el seguimiento de los procesos planeados a lo largo del tiempo.

Tarea	Descripción	Fecha inicio	Fecha fin	Duración (días)
1	Aprendizaje de programación con Qt GUI	01/04/2017	01/05/2017	30
2	Desarrollo de interfaz gráfica de usuario	25/04/2017	25/05/2017	30
3	Programación	28/05/2017	31/08/2017	95
3.1	Visualización de imágenes y vídeos	28/05/2017	18/06/2017	21
3.2	Administración y selección de etiquetas	19/06/2017	09/07/2017	20
3.3	Proceso de etiquetado	10/07/2017	07/08/2017	28
3.4	Exportación de Dataset	08/08/2017	15/08/2017	7
3.5	Almacenamiento en XML	16/08/2017	31/08/2017	15
4	Redacción de memoria	01/09/2017	26/09/2017	25
5	Preparación de la presentación	25/09/2017	05/10/2017	10

Figura 1 - Tiempos y fechas estimadas para cada tarea

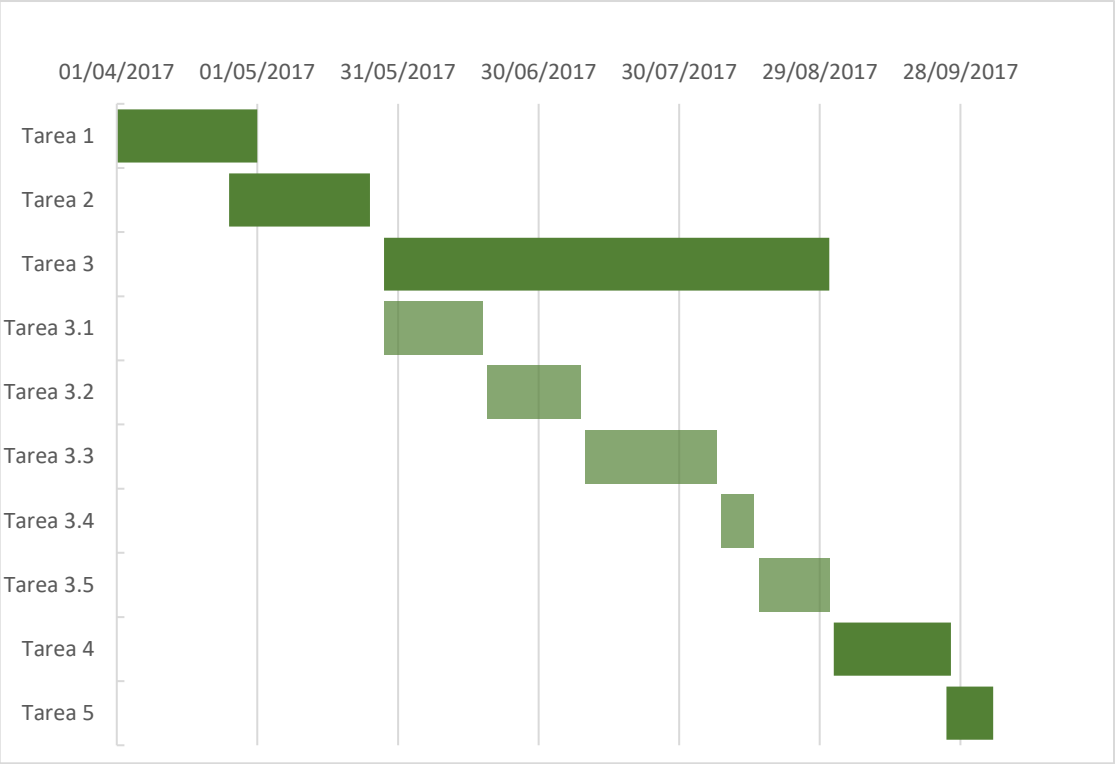

Figura 2 - Diagrama de Gantt para la planificación temporal del proyecto

1.3. ESTRUCTURA DEL DOCUMENTO

El presente documento se estructura en nueve capítulos con los siguientes contenidos:

- **Capítulo 1:** se hace una pequeña introducción sobre los vehículos autónomos y las bases de datos. Se acotan los objetivos establecidos para el desarrollo del proyecto y la planificación temporal del mismo.
- **Capítulo 2:** se describe el estado del desarrollo actual de los vehículos autónomos, los algoritmos de visión artificial para la conducción, así como las bases de datos más utilizadas y las herramientas de etiquetado para crearlas.
- **Capítulo 3:** se explica el contexto en el que se centra el desarrollo de la herramienta y los diferentes software empleados para ello.
- **Capítulo 4:** descripción detallada del software implementado para el desarrollo de la herramienta de etiquetado. Se describen las clases que se utilizan y las funciones principales del programa. Por último, se detalla sobre el proceso de

etiquetado y el almacenamiento de la información que realiza la herramienta, siguiendo los formatos establecidos por las bases de datos utilizadas.

- **Capítulo 5:** se ve un ejemplo de un frame etiquetado, con distintas etiquetas y varios polígonos, así como las salidas obtenidas al exportar el *Dataset*.
- **Capítulo 6:** se habla sobre la actual situación legislativa en cuanto a conducción autónoma en España, así como las líneas futuras del marco regulador que se deberá imponer.
- **Capítulo 7:** se describe el entorno socio-económico que rodea el proyecto, el posible futuro impacto que tendrá en la sociedad y el presupuesto estimado para el desarrollo del mismo.
- **Capítulo 8:** explicación de futuras implementaciones que se pueden realizar para mejorar y ampliar las posibilidades de la herramienta.
- **Capítulo 9:** se hace un análisis de los resultados obtenidos, así como una opinión personal sobre el desarrollo de la herramienta para concluir el presente trabajo.

2. ESTADO DEL ARTE

En esta sección se van a ver los elementos relacionados con la herramienta que se desarrolla en este proyecto y las alternativas comerciales que existen. Primero se introduce el concepto de vehículo autónomo y las compañías que están interesadas en su desarrollo y comercialización. Después se analiza la visión por computador enfocada hacia la conducción autónoma, así como alguno de los algoritmos más relevantes. A continuación, se van a ver las bases de datos existentes y su funcionamiento. Por último, se presentan las actuales herramientas para el etiquetado de imágenes que están disponibles en el mercado.

2.1. VEHÍCULOS AUTÓNOMOS

Un vehículo es considerado autónomo cuando es capaz de tomar el control de su propia conducción sin intervención humana, percibiendo el entorno que le rodea y navegando en consecuencia a la información procesada, para llegar al destino final que le ha indicado el ser humano [2].

Los vehículos autónomos son capaces de percibir su entorno gracias a una gran cantidad de elementos integrados en el mismo: láser, radar, GPS, lidar, cámaras de visión artificial, etc. Los sistemas de control del vehículo procesan la información con el objetivo de interpretarlos de la misma manera que lo haría un humano, buscando la ruta más apropiada para llegar a su destino, esquivando y reaccionando a los obstáculos que se encuentre por el camino.

Se han establecido unos niveles de autonomía de un vehículo, que sirven como meta para los desarrolladores e investigadores de este campo. Están divididos en seis niveles, del 0 al 5, en las que los tres primeros niveles precisan de la atención de un ser humano, mientras que los tres últimos necesitan un nivel de control humano menor [3]. Estos niveles pueden verse en la Figura 3.

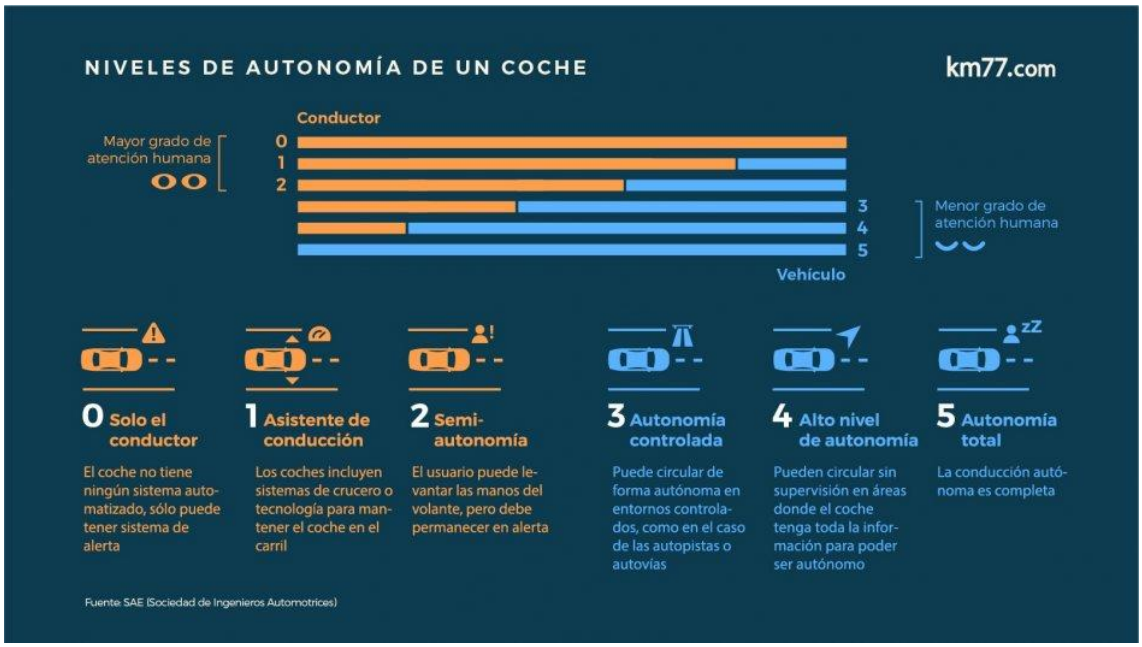

Figura 3 - Niveles de autonomía de un vehículo

Actualmente, muchas empresas se encuentras en vías de desarrollo de investigación sobre esta tecnología, ya que se espera que en 2020 existan vehículos totalmente autónomos circulando por las calles de todo el mundo. A continuación, se van explicar algunas de las propuestas que existen en el mercado del automóvil:

- **Audi**: la compañía alemana presentó en julio de 2017 su nueva serie, el A8, que se comercializará a partir de octubre de 2017. La característica por la que más destaca es que cuenta con un asistente de conducción en atascos de nivel 3, siendo el primero en alcanzar este nivel. También permite que el coche se aparque de forma autónoma a través de una aplicación *Android* y comparte información con otros vehículos de la marca, como los límites de velocidad o la presencia de un accidente de tráfico [4].
- **BMW**: gracias a su alianza con *Mobileye*, que suministrará los sistemas de percepción, e Intel, que aportará la computación de alto nivel, BMW ha puesto el año 2021 como fecha de lanzamiento del que dicen que será el primer vehículo autónomo de nivel 5. Este vehículo tendrá el volante como opción, ya que el usuario tan solo deberá introducir la dirección final. Esto será posible gracias al despliegue de las redes 5G y a la implementación del modelo de carreteras-colmena, donde todos los vehículos estén conectados

entre sí para comunicarse y compartir la información que permitirá un tráfico autorregulado, el fin de los atascos y la prevención de colisiones [5].

- **Mercedes**: la empresa germana presentó en 2015 el prototipo de su apuesta por los vehículos totalmente autónomos. Con los ojos puestos también en la siguiente década como fecha de lanzamiento, Mercedes apuesta por un vehículo híbrido de aspecto futurista y con un nivel de autonomía nivel 5, con asientos giratorios que permitan a los ocupantes del vehículo conversar mientras el sistema hace uso de los sensores, radares y cámaras que tendrá integrados para llevarlos a su destino final [6].
- **Google**: la compañía californiana se encontraba desarrollando desde 2014 un prototipo de vehículo biplaza llamado *Firefly* sin pedales ni volante que alcanzaba un máximo de 40 km/h. Sin embargo, ha decidido retirar el prototipo y apostar por una estrategia de desarrollo de las tecnologías de automatización, que podrán emplearse en vehículos ya disponibles comercialmente mediante alianzas con otras empresas [7].
- **Apple**: de forma similar a Google, el CEO de esta empresa confirmó en junio de 2017 que se encuentran desarrollando la tecnología necesaria para conseguir alcanzar la inteligencia artificial que controle a los vehículos autónomos, dejando a un lado el desarrollo de un vehículo suyo propio [8].

2.2. VISIÓN ARTIFICIAL

Dentro del campo de la Inteligencia Artificial, una de las disciplinas donde más avances se están consiguiendo es la llamada visión por computador o visión artificial. Se denomina así a los algoritmos que se realizan para el procesado, el análisis y la obtención de información de las imágenes, con el objetivo de saber interpretar los objetos, acciones u escenas que se encuentran en ellas.

La visión artificial se utiliza en un amplio número de campos, debido al gran desarrollo conseguido en los últimos años y a sus múltiples aplicaciones. Se utiliza tanto en el área de la seguridad con los reconocimientos faciales o los radares de las cámaras de tráfico, como en la Industria para la monitorización de procesos y el

control de calidad de los productos desarrollados. Cada vez se ven más aplicaciones para smartphones que utilizan esta tecnología: en redes sociales con filtros, como *Snapchat*, o como método de identificación biométrica empleada, por ejemplo, en el *Face ID* del *iPhone X*.

La visión por computador comenzó a desarrollarse en 1960, aunque hasta hace poco solo unos pocos especialistas podían utilizar estas técnicas, debido al alto coste de los programas que existían y la necesidad de potentes ordenadores [9]. Sin embargo, actualmente existen muchas más herramientas, como la librería *OpenCv*, que supone una solución de alto nivel y totalmente gratuita, explicada en el apartado 3.2.2.

Las imágenes que se procesan para extraer la información del entorno pueden ser archivos cargados, o imágenes en tiempo real tomadas desde cámaras. Antes, las cámaras empleadas eran grandes, caras y pesadas, pero en la actualidad pueden utilizarse versiones comerciales más baratas y ligeras, como las de un smartphone o la cámara *Kinect* que, aunque fue desarrollada por Microsoft para los videojuegos de su consola *Xbox 360*, supuso una revolución y un gran avance en los campos de la robótica y la conducción autónoma, ya que ofrece un sistema de reconocimiento del entorno 3D de alto nivel, con un coste reducido y de peso muy ligero.

Dentro del campo de la conducción autónoma, la visión por computador es una tecnología fundamental para la detección de objetos, vehículos, peatones, etc. La obtención de la información que tiene en su entorno un vehículo autónomo, así como conseguir clasificar correctamente dichos obstáculos, supone un requisito imprescindible para la toma de decisiones del sistema.

El proceso de toma de decisiones en un sistema automático como el de un vehículo autónomo necesita de múltiples y complejos algoritmos de aprendizaje. El *Deep Learning* [10] es muy utilizado en este campo de investigación, ya que resulta ser muy apropiado: usa conocimientos y modelos estadísticos para encontrar patrones, dentro de una serie de datos que aparentemente son aleatorios, para conseguir ser capaces de diferenciarlos automáticamente. El entorno que se

encuentra un vehículo autónomo es dinámico y muy variado, por lo que necesita encontrar estos patrones que permitan identificar los obstáculos, para así saber cómo actuar ante ellos.

Una de las maneras más comunes de implementar el *Deep Learning* el uso de redes neuronales profundas, que es una herramienta matemática empleada para modelar el funcionamiento de la red de neuronas del cerebro humano, que recibe una serie de números y los transforma mediante cálculos matemáticos en otro conjunto numérico que proporciona unos datos procesables para el ordenador.

El *Deep Learning* necesita de una gran cantidad de información para conseguir entrenar el algoritmo con un alto grado de precisión, por lo que es fundamental contar con buenas y amplias bases de datos, que aporten imágenes e información de calidad que permita entrenar sin introducir errores.

Para el procesamiento de objetos en el entorno gracias a la visión por computador se cuentan con varios algoritmos que enfocan en problema con distintas perspectivas y metodologías:

- La detección y clasificación de objetos busca las características que definen el elemento que están buscando, como puede ser el color, la textura, la forma, la orientación, etc. Por ejemplo se pude utilizar el algoritmo DPM [11] (*Deformable Part Model*), que trata de descomponer la compleja estructura de un objeto en pequeñas partes más sencillas de analizar y procesar.
- En la segmentación semántica el objetivo es asignar cada pixel de una imagen a una categoría de etiquetado, por ejemplo, a un color. Un algoritmo bastante empleado junto al *Deep Learning* es el de las redes neuronales convolucionales [11], que utilizan sus múltiples capas para la extracción de características de la imagen y después segmentar la información en función a dichas características.

Estos algoritmos son pre-entrenados con la base de datos *ImageNet*, la cual se explica en el siguiente apartado.

2.3. BASES DE DATOS

2.3.1 ImageNet

ImageNet [12] es un conjunto de datos de imágenes organizado por la jerarquía de *WordNet*. Esta base de datos es un recurso útil empleado por investigadores para entrenar redes neuronales profundas, ya que se puede identificar cada detalle de una imagen almacenada, y ser comparada a través de algoritmos de inteligencia artificial. Esta base de datos comenzó a desarrollarse debido a la necesidad de mejores datos para los campos de procesamiento de imágenes y visión por computador, por lo que es necesario un amplio conjunto de datos que permitan el entrenamiento de algoritmos más complejos y sofisticados.

Para conseguir el objetivo de una conducción autónoma eficiente y segura, son necesarias miles de imágenes para compararlas entre patrones y alcanzar un mejor resultado en el reconocimiento y la toma de decisión. *ImageNet* proporciona imágenes que son etiquetadas por humanos, que controlan la calidad y la utilidad de los datos administrados.

2.3.2 Pascal VOC

El proyecto *Pascal VOC* [13] (*Visual Object Classes*) es una base de datos de imágenes para la clasificación, detección y segmentación de objetos, así como para el reconocimiento de acciones. Recoge fotografías de consumo de la web *Flickr*, para realizarles anotaciones de alta calidad. Desarrollado entre 2005 y 2012, el proyecto VOC ha evolucionado hasta conseguir un gran conjunto de datos para la detección de objetos, la segmentación de instancias y el razonamiento por contexto. El conjunto de datos comprende 91 clases de objetos, 2.5 millones de instancias anotadas y 328 mil imágenes en total.

2.3.3 Cityscapes

Cityscapes [14] es una base de datos que se centra en el etiquetado de imágenes de calles urbanas, el almacenamiento de las mismas y los diseños 3D

de los patrones que se encuentra en ellas. Estas imágenes son capturadas en secuencias de vídeos que son grabados durante la conducción por las calles de múltiples ciudades del mundo.

Esta base de datos proporciona más de 30 clases distintas para el procesamiento de imágenes, con las que más de cinco mil imágenes están etiquetadas de forma fina o precisa, y otras veinte mil cuentan con etiquetado poco preciso. Además, esta plataforma cuenta con un conjunto de aplicaciones en las que subir las imágenes etiquetadas y obtener una clasificación por la comparación con la base de datos. Esto denominado *Benchmark Suite.*

Cityscapes está disponible de forma gratuita para entidades académicas y no-académicas, pueden ser utilizados sin fines comerciales, para la enseñanza y para aplicaciones científicas.

2.3.4 KITTI

KITTI [15] es un proyecto de "*Karlsruhe Institute of Technology and Toyota Technological Institute of Chicago*", que desarrolla nuevos desafíos de la visión por computador del entorno del mundo real mediante comparativas o *benchmarks*. Se trata de un conjunto de herramientas que funciona como base de datos, cuyo trabajo consiste en el almacenamiento de datos de una cámara estéreo, detección de objetos 3D, posicionamiento 3D, odometría visual y el flujo óptico.

El conjunto de imágenes ofrecido corresponde a la ciudad de Karlsruhe, Alemania, en carreteras y autopistas y en zonas rurales. Estos datos pueden encontrarse dentro de la Web oficial clasificados en distintas categorías que existen. En la última sección, *raw data*, se pueden descargar distintos tipos de datos etiquetados y no etiquetados para hacer pruebas.

No está permitido el uso comercial de esta base de datos, únicamente puede ser empleada para uso académico o de investigación. Junto a Cityscapes, estas bases de datos son las más utilizadas por el Laboratorio de Sistemas Inteligentes

de la Universidad Carlos III de Madrid, para el cuál se desarrolla este trabajo de fin de grado.

2.4. HERRAMIENTAS DE ETIQUETADO

2.4.1 LabelImg

Esta herramienta está implementada en lenguaje Python, y puede utilizarse en los sistemas operativos Windows y Linux. Se trata de una herramienta de etiquetado de imágenes, capaz de identificar en una imagen a una persona, animal, objeto, entre otros. Utiliza recuadros que el usuario dimensiona y posiciona en la imagen manualmente, con la idea de ser procesados posteriormente como *BoundingBoxes.*

LabelImg [16] tiene una licencia MIT, es decir, una licencia de Software libre que permite la reutilización con pocas restricciones. El etiquetado se almacena en documentos XML siguiendo el formato de *Pascal VOC*, con la idea de ser utilizado en la base de datos *ImageNet* [12].

2.4.2 LabelMe

Se trata de una herramienta online implementada utilizando los lenguajes JavaScript, CSS y HTML. *LabelMe* [17] es una herramienta de etiquetado basada en JavaScript para servidores (*online*), pudiendo acceder desde cualquier lugar gracias al almacenamiento en la nube. También cuenta con una aplicación para *smartphones* de sistema iOS.

El usuario puede realizar el etiquetado mediante puntos en la imagen, lo que permite un etiquetado más fiel a la forma original del objeto, persona, vehículo, animal, etc. Estos puntos marcarán el contorno, y el usuario debe introducir el nombre del objeto y los atributos que lo componen. Además, únicamente se admiten imágenes con extensión "*.jpg*".

2.4.3 LEAR Image Annotation Tool

LEAR Image Annotation Tool [18] está implementada en C++, para la cual es necesario tener instalado el programa *Qt Creator* y la librería de *OpenCV*, lo que supone que puede utilizarse en Windows, Linux y macOS.

Esta herramienta de anotación de imágenes mediante la creación de recuadros (*BoundingBoxes*) para el etiquetado, y también puede identificar la rotación de un objeto y usar puntos de fijación dentro de un objeto. Incluye opciones para modificar, editar, copiar y pegar los cuadros delimitadores, utilizar zoom y alinear o borrar los objetos seleccionados.

2.4.4 Ratsnake Image Annotation Tool

Se trata de una herramienta implementada en Java, considerada actualmente como uno de los mejores software para hacer etiquetado de imágenes. Puede utilizarse en los sistemas operativos Windows y Linux.

Con *Ratsnake* [19] se pueden hacer anotaciones semiautomáticas de secuencia de imágenes, segmentar rápido las imágenes en mallas o polígonos (o ambos) y exportar las anotaciones a otras aplicaciones como, por ejemplo, en un documento XML compatible con *LabelMe*. El etiquetado se realiza por puntos, lo que permite seleccionar la forma deseada de cada objeto.

2.4.5 RectLabel

RectLabel [20] es una herramienta implementada en C++ que puede obtenerse como aplicación en los dispositivos con sistema operativo macOS. Esta herramienta tiene soporte para crear salidas en formato de *Pascal VOC* y exporta los datos del etiquetado en formato *JSON*.

Permite el etiquetado de imágenes mediante rectángulos delimitadores. Es una de las herramientas más avanzadas, ya que permite la personalización de los atributos de las etiquetas con anterioridad, así como la carga de etiquetas de otras imágenes, lo que permite agilizar el proceso del etiquetado.

2.4.6 VGG Image Annotator

Se trata de una herramienta online implementada utilizando los lenguajes HTML, CSS y JavaScript. *VGG Image Annotator* [21] es un proyecto de Software Libre publicado bajo licencia *BSD-2*, por lo que permite la utilización de la herramienta con pocas restricciones. Puede utilizarse desde cualquier buscador moderno, aunque también existe una versión descargable para usar la herramienta offline.

El usuario puede realizar el etiquetado mediante formas predefinidas, como rectángulos, círculos o elipses, así como por puntos en la imagen que formen un polígono. Esto que permite un etiquetado más rápido o más preciso respecto a la forma original de un objeto, persona, vehículo, animal, etc, , en función a las necesidades del usuario. Permite la clasificación de lo etiquetado con la definición por parte del usuario del nombre y el color de la etiqueta, aunque también puede añadir otros atributos si así lo desea. Todo ello hace que sea una herramienta de segmentación de imágenes muy completa y útil.

3. DESCRIPCIÓN GENERAL DEL PROYECTO

En este apartado se va a explicar el contexto en el cuál se tomó la decisión de que era necesario desarrollar una nueva herramienta de etiquetado que se adaptase a las necesidades del programa de investigación de vehículos autónomos de la Universidad Carlos III de Madrid.

Además, se va a hablar de los softwares empleados para implementar dicha herramienta, sus características y la razón por la que se escogieron dichos recursos para el desarrollo de este proyecto.

3.1. CONTEXTO

Actualmente, existen proyectos de investigación para desarrollar vehículos inteligentes en el *Intelligent Systems Laboratory* de la Universidad Carlos III de Madrid, localizado en el campus de la Escuela Politécnica Superior de Leganés.

Estos vehículos utilizan sensores y actuadores específicos que emplean para la conducción autónoma, con el objetivo de navegar sin necesidad de monitorización humana por su entorno sin suponer un peligro para otros vehículos, peatones o el mismo vehículo.

Para ello, uno de los puntos clave es que cada vehículo cuente con un sistema de visión artificial preciso, que les permita funcionar de forma correcta en las distintas situaciones y entornos que pueda encontrarse durante la conducción. Este sistema de visión artificial necesita de algoritmos de aprendizaje que mejoren su capacidad de procesamiento de imágenes y clasificación de obstáculos que deriven, posteriormente, en una mejor toma de decisiones durante la conducción.

Para entrenar dichos algoritmos de aprendizaje, son necesarias bases de datos que ofrezcan un alto número de ejemplos, cumpliendo con las necesidades exigidas por los algoritmos. Las bases de datos más utilizadas en el Laboratorio son *Cityscapes* y *KITTI* (apartados 2.3.3 y 2.3.4, respectivamente), que ofrecen una gran

variedad de imágenes anotadas y etiquetadas para el procesamiento por visión artificial y detección de objetos.

Aunque esto es aplicable a todos los vehículos, desde turismos a drones, en este proyecto vamos a centrarnos en dos de las plataformas de investigación en las que se trabaja actualmente el laboratorio: el *iCab* y el *IVVI 2.0*.

Ambos necesitan de buenas bases de datos que les permitan desarrollar y entrenar sus sistemas autónomos de conducción. Dichas bases se conforman de imágenes previamente etiquetadas, que utilizan para entrar los algoritmos. Este proyecto desarrolla una herramienta de etiquetado que sea sencilla e intuitiva que facilite el trabajo de anotar las imágenes del investigador del Laboratorio, estando además adaptada a las necesidades de los vehículos desarrollados por el LSI.

3.1.1 iCab

El *Intelligent Campus Automobile*, también conocido como iCab [22], es la denominación para un tipo de vehículo desarrollado por el Laboratorio. Se trata de un UGV, siglas de *Unmanned Ground Vehicle*, es decir, vehículo terrestre no tripulado. Actualmente, el Laboratorio cuanta con dos carritos de golf eléctricos que se mueven de forma autónoma en el campus de Leganés, con la función de transportar a miembros de la Universidad o visitantes entre diferentes puntos de los edificios del campus.

Estos vehículos han sido transformados tanto mecánica como electrónicamente. Obtienen información del entorno operativo gracias a los múltiples sensores con los que está equipado, que les permiten moverse por el campus sin la ayuda humana durante un tiempo prolongado y con el objetivo de no provocar situaciones que puedan entrañar un peligro para el entorno, los humanos o el propio vehículo. Además, cada vehículo cuenta con un módulo inalámbrico para comunicarse entre ellos y compartir la información del entorno.

El sistema de los iCab es capaz de detectar y clasificar los objetos que se encuentra durante el recorriendo, tanto estáticos como puede ser un árbol o un edificio, como los dinámicos, como peatones, otros vehículos, animales, etc. Para ello, y debido a que opera dentro del campus de la Universidad, estos vehículos hacen uso de bases de datos específicamente creadas para ellos.

El control de los vehículos se consigue gracias a un ordenador incorporado a bordo, que utiliza algoritmos implementados en arquitectura ROS [23] (*Robot Operating System*), logrando un mejor rendimiento en la comunicación entre los procesos y los múltiples sensores.

Figura 4 - iCab 1 del Laboratorio de Sistemas Inteligentes de la UC3M

3.1.2 IVVI 2.0

La segunda plataforma de investigación desarrollada por el Laboratorio de Sistemas Inteligentes se denomina IVVI 2.0, acrónimo de *Intelligent Vehicle based on Visual Information* [24]. Este vehículo está equipado con los sensores y sistemas de procesado más avanzados en el desarrollo y las pruebas de los ADAS, siglas de *Advanced Driver Assistance Systems* (Sistemas Avanzados de Asistencia al Conductor).

Anterior a este vehículo, el Laboratorio desarrolló hasta el año 2009 la plataforma denominada IVVI 1.0 actualmente, siendo la primera plataforma experimental de investigación y desarrollo de ADAS del LSI, basándose en el análisis de imágenes y visión por ordenador. Los resultados de esta investigación se están implementando y ampliando en el vehículo de la plataforma IVVI 2.0.

Figura 5 - IVVI 2.0 del Laboratorio de Sistemas Inteligentes de la UC3M

En este vehículo, todos los ordenadores, sensores e interfaces con los que está equipado, se encuentran casi totalmente integrados en el turismo, con una baja influencia en el aspecto visual del coche. Esto hace del IVVI 2.0 un equipo de vanguardia, que se ha diseñado en función a estas tendencias en la asistencia a la conducción.

Gracias a los dispositivos que están integrados en el vehículo, entre muchas otras funciones, se pueden detectar por visión artificial vehículos u objetos del entorno, así como señales de tráfico que son clasificadas, todo bajo condiciones de conducción diurna. El IVVI 2.0 también cuenta con una cámara infrarroja que detecta el calor corporal de los peatones, siendo capaz de detectarlos en condiciones de baja o incluso nula luminosidad, o con un sensor del movimiento

que detecta el rostro del conductor, emitiendo una alarma si procesa que se está durmiendo.

El procesamiento de los datos de los sensores se realiza en una plataforma informática de arquitectura ROS, que permite la fusión de la información de alto y bajo nivel.

3.2. SOFTWARE UTILIZADO

3.2.1 Qt Creator

Qt Creator [30] es un entorno de desarrollo integrado multiplataforma que permite la implementación de Interfaces Gráficas de Usuario(*GUI*). Este programa se empieza a desarrollar en 1994 por la ahora conocida como *Qt Development Frameworks* (anteriormente conocida como *Trolltech*). Esta plataforma es una de las más utilizadas para el desarrollo de softwares aplicados en vehículos, automatización industrial, elementos médicos, etc. *Qt Creator* es un programa que se puede utilizar en los sistemas operativos Windows, Linux o macOs, haciendo a este entorno muy versátil para muchos tipos de aplicaciones.

Qt tiene código abierto, cuyas herramientas son gratuitas y, gracias a la comunidad de programadores y la financiación por parte de Nokia, existe un continuo desarrollo de la plataforma [26].

En este proyecto se ha utilizado el entorno en el sistema operativo Linux Ubuntu 16.04, utilizando el lenguaje de programación orientada a objetos, *C++. L*a elección de este lenguaje es debido al previo conocimiento para programar con él, obtenido durante los estudios universitarios, además de ser el más utilizado por los miembros del Laboratorio de Sistemas Inteligentes.

Con *Qt Creator* se ha desarrollado una Interfaz Gráfica de Usuario, utilizando las clases aportadas por el módulo *Qt GUI* [27]. Esta interfaz permite de forma intuitiva y visual que el usuario realice el etiquetado de las imágenes, pudiendo crear las etiquetas con los atributos que él quiera, las formas tan precisas como

considere necesario y facilitándole la navegación por un vídeo o grupo de imágenes tan rápido o tan lento como desee.

3.2.2 OpenCV

OpenCV [28] (*Open Source Computer Vision*) es una librería libre empleada para la programación de códigos para la visión artificial, que fue desarrollada originalmente por Intel en 1999. Esta librería se utiliza para una infinidad de aplicaciones distintas, desde el control de calidad de los productos creados en una fábrica industrial, hasta sistemas de seguridad por detección de movimiento o reconocimiento facial. *OpenCV* es un programa que se puede utilizar en los sistemas operativos Windows, Linux o macOs. Está programada en C y C++, aunque también admite ser utilizada con los lenguajes Java y Phyton. Además, es un código libre y gratuito, que cuenta con la licencia BSD, la cual permite utilizar esta librería tanto con fines comerciales como académicos o de investigación. Todo ello, hace de esta biblioteca muy útil para diferentes tipos de aplicaciones y programadores.

Cuenta con más de 2500 funciones implementadas, de mayor y menor complejidad, que se adaptan a los proyecto más pequeños y a las programaciones de mayor nivel.

En este proyecto se ha utilizado el entorno en el sistema operativo Linux Ubuntu 16.04, utilizando el lenguaje de programación orientada a objetos, *C++,* y en conjunto con el programa *Qt Creator*. La elección de utilizar esta librería es debido al previo conocimiento para implementarla, obtenido durante los estudios universitarios, además de ser el muy utilizado por los miembros del Laboratorio de Sistemas Inteligentes y resultar útil y sencilla para conseguir el procesado de imagen necesario.

Con *OpenCV* se ha desarrollado todas las funciones relativas al procesado de la imagen, como la desencriptación de los frames de un vídeo para poder mostrarlos, la representación gráfica de los polígonos etiquetados sobre la

imagen o la creación de las imágenes etiquetadas para las salidas del *Dataset* del programa. Las clases de este programa más utilizadas han sido *Mat*, como matriz que representa una imagen, *Point*, que almacena las coordenadas de un punto, y *VideoCapture* para la utilización en frames de un vídeo.

3.2.3 TinyXML-2

TinyXML-2 [29R] es una librería libre y ligera que analiza un documento XML y crea a partir de él un DOM (*Document Object Model*) que puede leer, modificar y guardar. Fue creado por Lee Thomason y es comúnmente utilizado para el almacenamiento de datos en serie en documentos en XML (*eXtensible Markup Language*), que es un lenguaje de marcado legible por una máquina, que define y describe los distintos atributos de los datos.

Utiliza menor asignación de memoria, además de ocupar menos espacio que su predecesor, siendo cinco veces más rápido en la lectura del documento. Esta librería está implementada en lenguaje C++, teniendo su propio espacio de nombres (*namespace*). Está diseñado para ser sencillo y rápido de aprender, además de ser muy fácil de añadir a los proyectos que se realicen.

4. IMPLEMENTACIÓN DEL SOFTWARE DE ETIQUETADO

En este capítulo se va a explicar la implementación del software desarrollado para la herramienta de etiquetado, explotando las áreas que componen esta herramienta, las clases de objetos de programación más importante, así como la descripción de las funciones más importantes del programa. Por último, se verá con detalle los dos procesos más importantes de la herramienta: el Etiquetado y el almacenamiento de la información relativa al mismo.

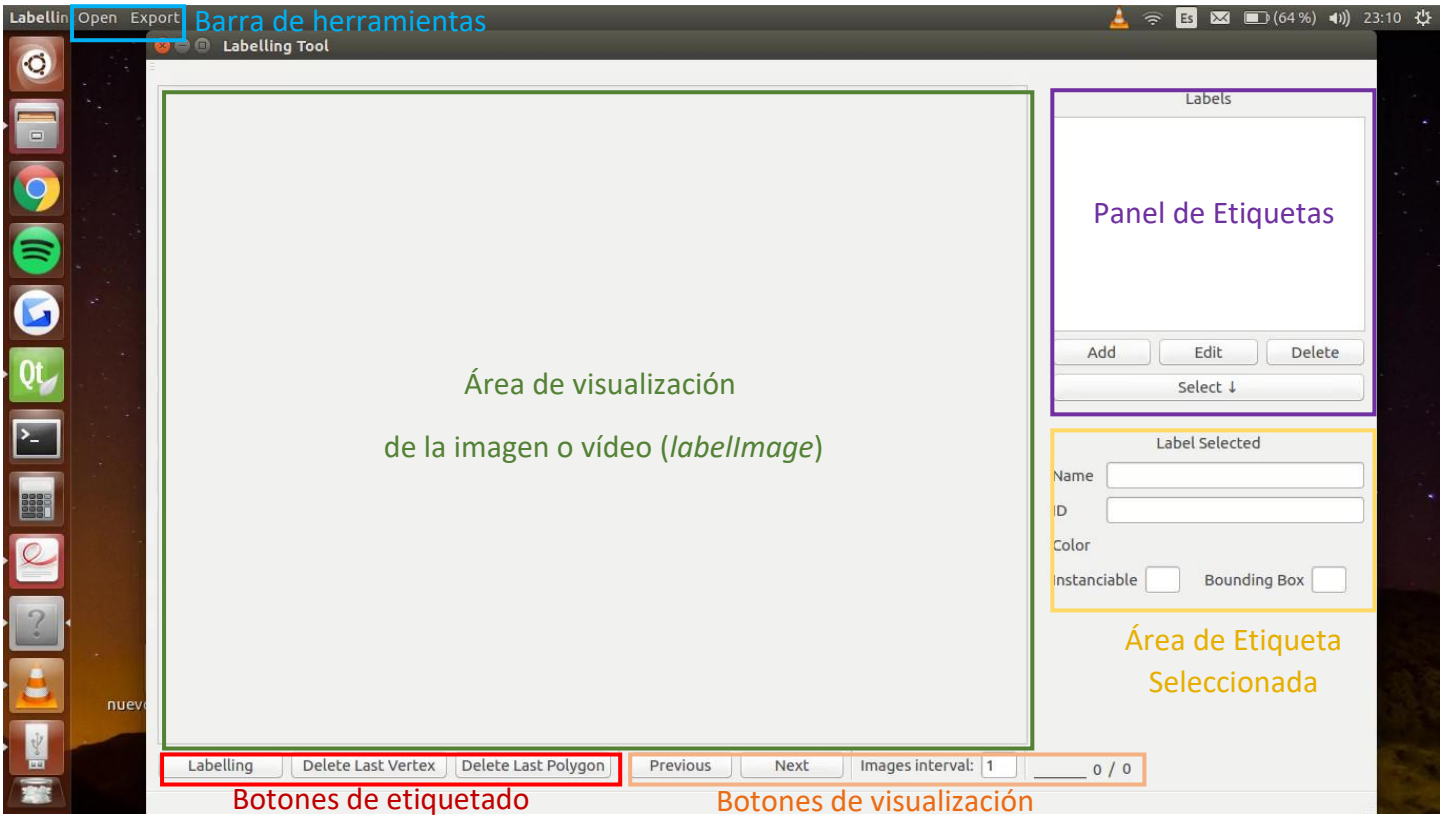

Figura 6 - Áreas de la ventana principal de la herramienta de etiquetado

En la Figura 6 se pueden observar las áreas que componen la ventana principal del programa, a las cuales se les ha asignado un nombre, que será utilizado durante las explicaciones de este capítulo.

4.1. CLASES

En este apartado se van a explicar las clases de objetos de programación más utilizados. En la programación del código del programa se han utilizado muchos tipos de objetos, sobre todo de las librerías de *Qt Creator*, *OpenCV* y *TinyXML-2*, pero en ese apartado únicamente se van a exponer las clases más importantes, es decir, las de los elementos utilizados por el usuario en la herramienta. En la Figura 7 se puede ver el UML que representa las relaciones que existen entre las clases.

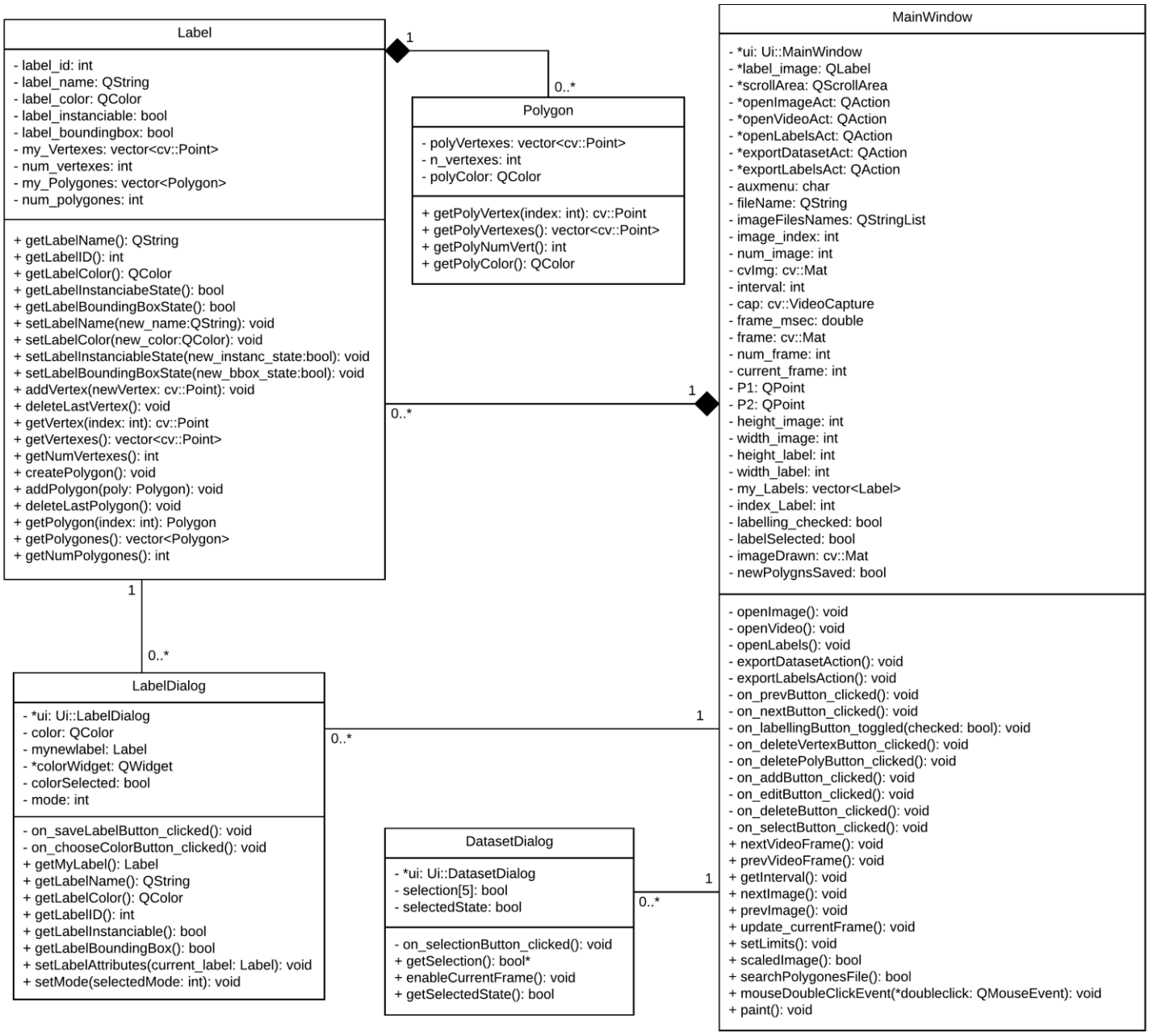

Figura 7 - Diagrama de clases de la herramienta de etiquetado

4.1.1 MainWindow

Se trata de la clase principal de la herramienta, ya que contiene toda la información que se va creando, así como la interfaz gráfica que se muestra como ventana principal.

En ella se guardan como atributos las acciones relativas a las acciones como abrir archivos o guardar la información, así como variables que facilitan la utilización de la información entre distintas funciones para el correcto funcionamiento del programa. Cabe destacar que en esta clase existe un contenedor que guardar todas las etiquetas creadas o cargadas por el usuario, así como el tamaño y las coordenadas de la imagen o frame respecto al sistema de referencia del MainWindow.

En esta clase están definidas las funciones que realizan la acción de abrir y cargar la información de un grupo de imágenes, un vídeo o un archivo XML que contiene etiquetas. También se definen las acciones de los botones presentes en la interfaz gráfica, así como la creación de los vértices o los polígonos, que se explicarán con mayor detalle en el apartado 4.3.

4.1.2 Label

Esta clase representa las etiquetas que se utilizan en la herramienta para el etiquetado de las imágenes. En ella se guardan las propiedades de las mismas, que son: el nombre, el ID (que representa el nivel de gris asociado a la etiqueta), el color con el que se dibujarán los polígonos, si se trata de un objeto instanciable y si se asocian *BoundingBox* a los polígonos de esta etiqueta. Por otro lado, también tiene dos contenedores donde se guardan los vértices y los Polígonos creados por el usuario.

En ella están definidas funciones para enviar los atributos de la etiqueta, así como para establecerlos de forma individual. También tiene funciones que administran los contenedores de los vértices y los polígonos, añadiendo nuevos,

borrando el último de ellos, enviando la información de uno de estos objetos o alguno de los contenedores entero.

4.1.3 Vertex

Esta clase no está definida implícitamente, ya que se utiliza la clase *Point* [30] de la librería OpenCV para guardar las coordenadas de los vértices que componen un polígono, o los que va marcando el usuario para componer un polígono. Estas coordenadas están calculadas respecto al sistema de referencia de la imagen o frame, que se encuentra en la esquina superior izquierda. El cálculo de dichas coordenadas se define en el apartado 4.3.3 Creación de un vértice.

4.1.4 Polygon

Esta clase representa a los polígonos, y tiene por atributos los vértices que lo componen y el color de la etiqueta a la que está asociado. Podemos enviar los valores de estas características a otra función gracias a los métodos que tiene definidos esta clase. El proceso de creación de estos objetos se explica en el apartado 4.3.4 Creación de un polígono.

4.1.5 LabelDialog

Cuando se quiera añadir una nueva etiqueta, o editar una ya existente, se abre una interfaz gráfica auxiliar en la que se eligen los atributos deseados para dicha etiqueta. Esta ventana tiene los parámetros vacíos en caso de seleccionar “Add” en el Panel de Etiquetas o, si se pulsa “Edit”, los atributos de la etiqueta que se desee editar.

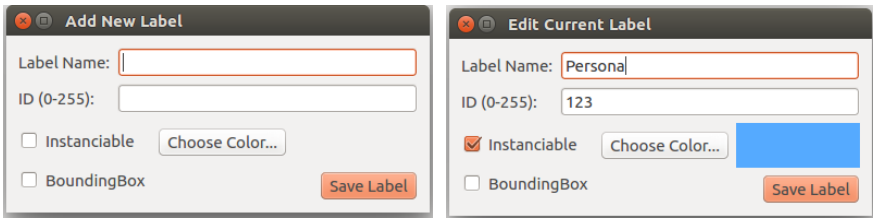

Figura 8 - Tipos de diálogos de la clase LabelDialog

En esta clase se definen todos los parámetros necesarios para adquirir la información aportada en ese diálogo, con los que se crea una nueva *Label* que es enviada para ser añadida al contenedor de la *MainWindow*.

4.1.6 DatasetDialog

Para guardar toda la información de una imagen ya etiquetada, existen diferentes salidas que conforman el *Dataset* de la misma. Las distintas opciones pueden elegirse por el usuario según sus necesidades en una interfaz gráfica emergente que se muestra al seleccionar esta opción. Esta clase almacena dicha elección de opciones, para después enviarla al *MainWindow* y así generar el *Dataset* solicitado.

4.2. FUNCIONES PRINCIPALES

En este apartado se va a explicar la funcionalidad de los botones presentes en la interfaz gráfica del usuario, así como las opciones situadas en las pestañas de la barra de herramientas superior de la ventana principal.

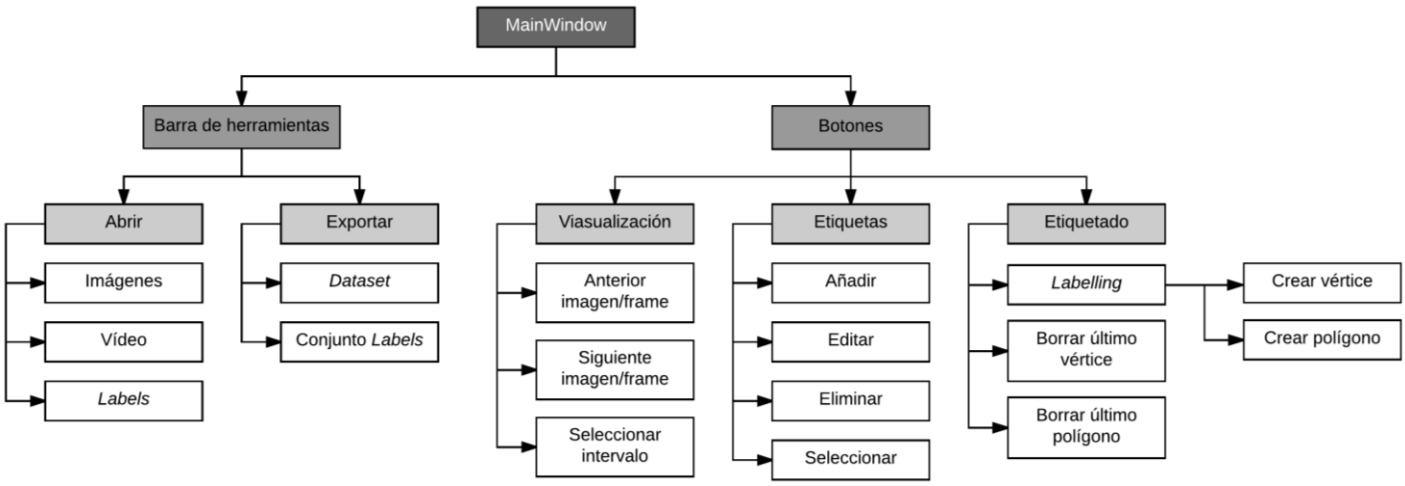

Figura 9 - Esquema de las funciones principales

4.2.1 Pestaña "Open"

En la Figura 10 se muestran las opciones existentes para la primera pestaña de la Barra de Herramientas.

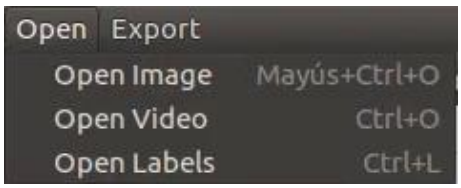

Figura 10 - Opciones de la pestaña Open de la Barra de herramientas

- **Abrir Imágenes:**

  Con la opción "*Open Image*", se abre un diálogo emergente que permite al usuario seleccionar las imágenes que quiera etiquetar, siendo posible cargar más de una imagen a la vez. Los botones de *Next* y *Previous* permiten navegar entre las distintas imágenes cargadas.

  Para este tipo de entrada, se admiten los dos formatos de imagen más usuales: la extensión *.jpg* (*Joint Photographic Experts Group*) y los *.png* (*Portable Network Graphics*).

- **Abrir Vídeo:**

  Al seleccionar la opción "*Open Video*", se muestra al usuario un diálogo emergente que le permite seleccionar el vídeo que quiere etiquetar, siendo posible ver frame a frame el vídeo, utilizando los botones de *Previous* y *Next* para ir hacia delate y hacia atrás en el vídeo, respectivamente.

  Con esta entrada, se permiten cargar los dos formatos de vídeo más utilizados por el departamento: la extensión *.avi* (*Audio Video Interleave*) y los *.mpg* (*Moving Picture Experts Group*).

- **Abrir Etiquetas:**

  Esta última opción, llamada "*Open Labels*", se muestra al usuario un diálogo emergente que le permite seleccionar un documento XML en el que esté almacenada la información relativa a una o más etiquetas. Estos Labels

cargados se añadirán al Panel de Etiquetas (ver Figura 6), pudiendo así reutilizar etiquetas que se usen con frecuencia, sin ser necesario crearlas cada vez que se ejecuta la herramienta de etiquetado.

Esta entrada permiten cargar documentos con la extensión *.xml* (eXtensible Markup Language), los cuales contienen una estructura de almacenamiento que se explicará con más detalle en el apartado 4.4.2 Documentos XML.

4.2.2 Pestaña "Export"

Las opciones presentes en la segunda pestaña de la Barra de Herramientas se muestran en la Figura 11.

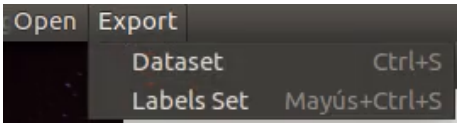

Figura 11 - Opciones de la pestaña Export de la Barra de herramientas

Cabe destacar que ambas opciones son explicadas de forma genérica en este apartado, y con mayor detalle y precisión en cuanto al cómo se guarda esta información en el apartado 4.4.

- **Exportar *Dataset*:**

  Esta opción, llamada "*Export Dataset*", abre un diálogo que permite al usuario seleccionar las salidas que quiere exportar del etiquetado que ha realizado. De este diálogo, se envía al *MainWindow* qué opciones han sido seleccionadas para realizar únicamente las operaciones requeridas por el usuario.

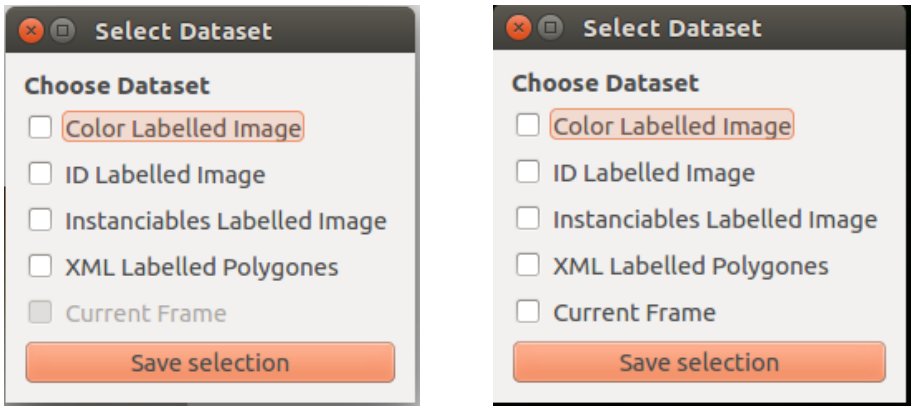

Figura 12 - Diálogo de selección del Dataset (imagen y vídeo)

Las salidas que el usuario puede seleccionar son las siguientes:

» *Color Labelled Image*: imagen definida con las dimensiones de la original, en la que se pintan los polígonos creados con el color asociado a la etiqueta a la que pertenece cada polígono.

» *ID Labelled Image*: imagen en la que se dibujan los polígonos creados durante el etiquetado, rellenados con el nivel de gris (propiedad ID) seleccionado para la etiqueta a la que están asociados cada uno de ellos.

» *Instanciables Labelled Image*: de forma similar a la salida anterior, en este caso únicamente se dibujan los polígonos de las etiquetas que tienen activa la propiedad de ser instanciables, siendo rellenado con el nivel de gris correspondiente.

» *XML Labelled Polygones*: esta salida genera un documento XML que guarda toda la información relacionada con los polígonos creados durante el etiquetado de la imagen. Esta forma de almacenamiento se explica con más detalle en el apartado 4.4.2 Documentos XML.

» *Current Frame*: esta opción únicamente está activa en caso de estar etiquetando un vídeo, ya que permite guardar como imagen el frame en el que se está etiquetando.

- **Exportar conjunto de *Labels*:**

  La segunda opción, denominada "*Export Labels Set*", abre un diálogo que permite al usuario exportar las etiquetas que se encuentran almacenadas en el Área de Etiquetas (ver Figura 6). Se pueden elegir el nombre y la ubicación del documento XML, en el que se van a almacenar todos los atributos de los Labels, con una estructura que se explicará con mayor detalle en el apartado 4.4.2 Documentos XML, mencionado anteriormente.

4.2.3 Botones de visualización

En la Figura 13 se muestran los Botones de visualización. Se encuentran en el margen inferior de la ventana principal, en la zona central y debajo del Área para la visualización de imagen y vídeo.

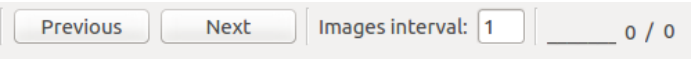

Figura 13 - Botones de visualización

- **Anterior:**

  El botón "*Previous*" permite retroceder a frames anteriores, en caso de que se esté etiquetando un vídeo, o navegar hacia atrás dentro del grupo de las imágenes cargadas por el usuario.

  Al pulsar este botón, la herramienta retrocedece tantos frames o imágenes como indique el número ubicado en el recuadro de "*Images Interval*". Debido al alto número de frames que suelen contener los vídeos, o si se abre un gran número de imágenes" esta funcionalidad permite al usuario la navegación más rápida por los elementos cargados de forma sencilla e intuitiva.

- **Siguiente:**

  Para realizar la operación contraria, es decir, avanzar hacia delante, tanto en los frames de un vídeo como en las imágenes cargadas por el usuario, se utiliza el botón "*Next*".

Del mismo modo que el botón anterior, al pulsar *Next*, el programa avanzará el número de frames o imágenes que se haya seleccionado en el recuadro de "*Images Interval*", facilitando y simplificando la navegación del usuario por los elementos a etiquetar.

Si se pulsa este botón, o el botón "*Previous*", y existen nuevos polígonos etiquetados que no han sido guardados, la herramienta pregunta al usuario si desea realmente cambiar la imagen o el frame. Si se cambia, todos los polígonos se borran de los contenedores de las etiquetas, por lo que se ha implementado esta barrera de seguridad para evitar que el usuario pierda los datos etiquetados si pulsa uno de estos botones por error.

- **Selección de intervalo:**

  Como ya se ha explicado, en este recuadro asociado al texto "*Images Interval*" el usuario puede indicar cuántos frames en un vídeo o imágenes de las cargadas quiere avanzar o retroceder.

  Cada vez que se pulsa *Next* o *Previous,* el software comprueba el número escrito en ese momento en el recuadro, por lo que el usuario puede modificar este valor cuando lo considere necesario. Esto le permite navegar por los elementos abiertos con mayor rapidez o con mayor precisión, dependiendo de las necesidades de ese momento.

- **Imagen o frame actual:**

  A la derecha de la Selección de intervalo, encontramos un contador que indica en qué imagen o frame estamos etiquetando.

  De esta manera, cuando se abren un grupo de imágenes, en esta zona aparece la palabra "*Image*", seguido de la posición de la imagen que está abierta en ese momento dentro del grupo que hemos abierto y, después de la barra inclinada, el número total de imágenes. De forma similar, al abrir un

vídeo, se muestra el texto "*Frame*" seguido del número del fotograma que está abierto y el número total de frames con los que cuenta el vídeo.

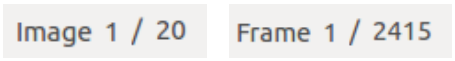

Figura 14 - Contador de la imagen o el frame actual

4.2.4 Panel de Etiquetas

En la Figura 15 se muestran los elementos del Panel de Etiquetas, que se encuentra en la parte superior derecha de la ventana principal.

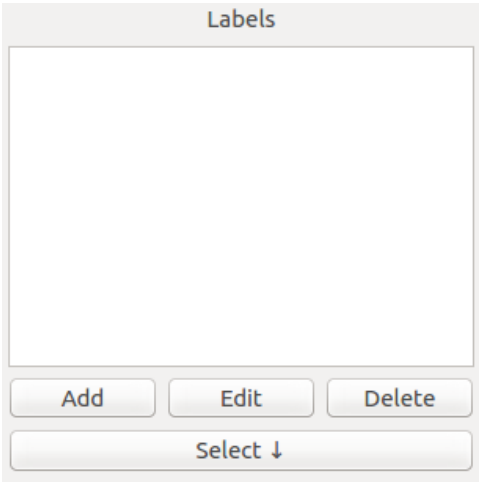

Figura 15 - Botones del Panel de Etiquetas

- **Lista de etiquetas:**

  Cuadro que lista los nombre de las *Labels* que existen para el etiquetado. El orden en el que se encuentran en la lista corresponde con el orden en el que están en el contenedor de etiquetas de la *MainWindow,* lo que facilita las operaciones entre la lista y el contenedor.

  Al añadir una etiqueta, el nombre de ésta se añade al final de la lista. Para editar, eliminar o seleccionar una etiqueta, es necesario elegir el elemento de la lista con el nombre de la misma.

- **Añadir:**

El botón "*Add*" abre una interfaz gráfica auxiliar en la que se eligen los atributos deseados para la etiqueta que queremos crear y añadir al programa principal.

Este diálogo tiene los parámetros vacíos, que el usuario debe rellenar: el nombre, el ID (que representa el nivel de gris), el color que quiere relacionarse con la etiqueta, si se trata de un objeto instanciable y si se asocian *BoundingBox* a los polígonos creados a partir de este *Label*.

El software comprobará que cumple con las condiciones mínimas, como que el nombre no esté vacío, que el ID tenga un valor entre 0 y 255 o que el color no pueda ser negro, para que no quede invisible al crear las imágenes al general el *Dataset*, que se forman en un objeto *Mat* de fondo negro.

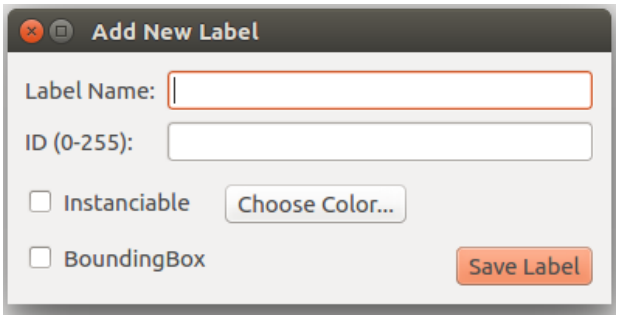

Figura 16 - Interfaz para añadir una nueva etiqueta

- **Edit:**

Con la opción del botón "*Edit*" se abre un diálogo emergente en el que se muestran los atributos de la etiqueta que queremos editar, para que el usuario modifique los que desea cambiar.

Si la etiqueta editada es la que se muestra en el Área de Etiqueta Seleccionada, esta zona se actualiza con los nuevos atributos. Esto se hace para que no quede en esa área una información anterior, que pueda ocasionar confusión al usuario. También se actualizan los dibujos de los polígonos asociados a esta etiqueta, mostrando las nuevas propiedades.

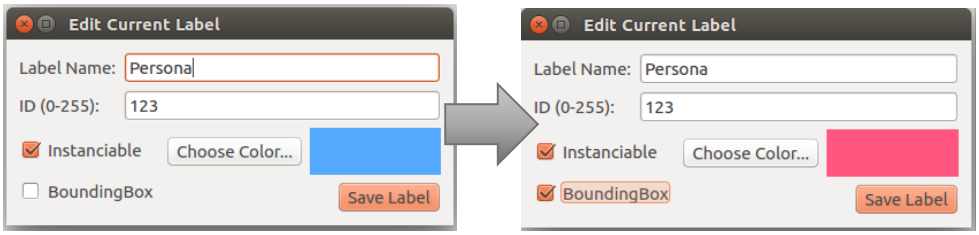

Figura 17 - Edición de las propiedades de un Label

- **Eliminar:**

Para borrar un *Label*, primero se debe seleccionar en la Lista de Etiquetas. Con la posición en esta lista, se accede a la etiqueta del contenedor de *Labels* del programa para eliminarla y, posteriormente, ese elemento en la Lista de Etiquetas.

Con esto, también ha de actualizarse la imagen etiquetada, ya que todos los polígonos y los vértices asociados a la etiqueta borrada se destruyen con ella. Por ello, es necesario volver a dibujar, para obtener una imagen en la que esta etiqueta ya no esté representada.

- **Seleccionar:**

Durante el proceso de etiquetado, es necesario tener un *Label* seleccionado. Éste es el que aporta las propiedades de dibujo a los vértices y los polígonos, y el que almacena ambos tipos de objetos en sus contenedores.

Tras elegir el *Label* deseado en la Lista de etiquetas, al pulsar el botón "*Select*", las propiedades de la etiqueta se mostrarán en el Área de Etiqueta Seleccionada, facilitando de forma visual y sencilla el conocimiento de qué valores se están utilizando en cada momento para el usuario.

Si la etiqueta que se encuentra mostrada en esta área es modificada con el botón *Edit*, los valores se actualizarían automáticamente. Esto ayuda a evitar que el usuario crea que está etiquetando con unas propiedades de la etiqueta que ya no son las reales. De forma similar, si la etiqueta es

eliminada de la lista con la opción *Delete*, el Área de Etiqueta Seleccionada se reinicia para mostrar todos los parámetros vacíos.

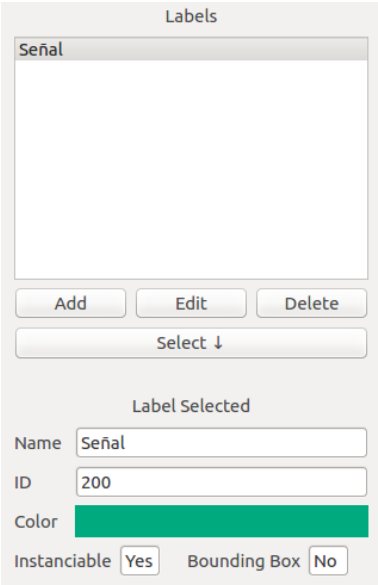

Figura 18 - Selección del Label para el etiquetado

4.2.5 Botones de etiquetado

La Figura 19 muestra los elementos que componen los Botones para el etiquetado, que se encuentran en el margen inferior de la ventana principal, debajo del área para la visualización de imágenes y vídeo, en la zona izquierda.

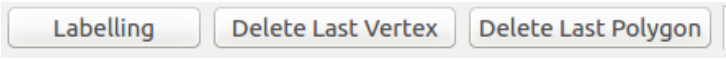

Figura 19 - Botones para el etiquetado

- ***Labelling***:

  Este botón representa la propiedad de estar en modo “etiquetando”, teniendo un estado activado y otro desactivado. Con esto, el usuario comunica al programa que las selecciones que haga sobre la imagen en ese momento será con intención de crear vértices o polígonos nuevos. Así se evita que se creen nuevos elementos por error cuando el usuario no tenga intención de etiquetar.

  La creación de vértices y polígonos están explicados con mayor detalle en los apartados 4.3.3 y 4.3.4, respectivamente.

- **Borrar último vértice:**

Con este botón, llamado "*Delete Last Vertex*", el usuario puede eliminar el último vértice del contenedor de la etiqueta que está seleccionada. Es decir, que se pueden borrar tantos vértices como se desee, siempre en el orden inverso al que se han creado.

Esto permite la corrección de errores, tanto en la creación como en el posicionamiento de los vértices, y de una forma intuitiva y sencilla para el usuario. Es importante entender que se eliminan los vértices de la etiqueta que está seleccionada en ese momento y, que al crear un polígono, los vértices son eliminados del *Label* para crear dicho *Polygon*, por lo que no es posible eliminarlos uno a uno. En caso de que el usuario haya creado el polígono y quiera cambiar un vértice, deberá eliminar el Polygon entero.

- **Borrar último polígono:**

De forma similar a la opción anterior, el usuario cuenta con otro botón denominado "*Delete Last Polygon*". Éste permite al usuario borrar el último polígono del contenedor de la etiqueta que está seleccionada. Es decir, que es posible borrar tantos polígonos como se desee, de distintas etiquetas: solo hace falta seleccionar la etiqueta asociada al polígono, y se van eliminando de en el orden inverso al que se han creado.

De nuevo, esto facilita la corrección de errores para el usuario de forma intuitiva y simple, tanto en si se ha producido el error en la creación del polígono como si no tiene la forma deseada finalmente.

Tanto en esta opción, como en la anterior, cuando un polígono o un vértice son eliminados, es necesario actualizar la imagen dibujada que se está mostrando, para que recoja los cambios sufridos tras la eliminación de un elemento.

4.3. ETIQUETADO DE IMÁGENES

En este apartado se va a explicar la lógica y la funcionalidad del proceso de etiquetado de una imagen, explicando primero de forma genérica las condiciones y el método utilizado, para después ampliar las funciones más importantes.

Para poder etiquetar, el usuario debe hacer doble click con el botón izquierdo del ratón donde quiera situar un nuevo vértice del polígono que quiera crear. Esta función está implementada con la acción de doble pulsación para evitar la creación de falsos vértices, ya que es más común pulsar una vez con el ratón por error, que haciendo doble click. Sin embargo, para mayor control de las intenciones del usuario, éste debe activar el botón *Labelling* mientras etiquete. Si se cometiese algún error, o el usuario quisiera corregir algún vértice o polígono, puede hacer uso de los botones *Delete Last Vertex* y *Delete Last Polygon*, respectivamente.

Una vez el usuario realiza el doble click, el software debe comprobar que el punto donde el usuario está pulsando el ratón para crear un nuevo vértice está dentro de la imagen o el frame mostrado en ese momento. Para ello, es necesario calcular la posición de la figura para conocer los márgenes de la misma, que delimitarán la zona de etiquetado permitida.

Tras comprobar que el nuevo vértice es correcto, el software lo incluirá en el contenedor de Vertex de la etiqueta seleccionada y procederá a dibujar este vértice. Para ello, es necesario realizar una transformación de las coordenadas de la posición del ratón respecto al sistema de referencia de la ventana principal, a las coordenadas del vértice respecto al sistema de referencia de la imagen o vídeo.

Para crear un nuevo polígono, el usuario tan solo debe volver a pulsar en el primer vértice dibujado, para cerrar la forma. Se admite un pequeño margen de ±10 píxeles en horizontal y vertical, ya que es bastante complejo pulsar exactamente en las coordenadas del primer vértice. Sin embargo, antes de crear el polígono, surge un diálogo que le preguntará si realmente quiere formar el objeto. Esto se hace por

si el usuario realmente quisiera colocar un vértice cercano al primero por motivos de precisión.

El proceso de etiquetado se ha implementado pensando siempre en conseguir un funcionamiento sencillo y cómodo de usar para el usuario, que le facilite la tarea para conseguir las salidas etiquetadas necesarias para las bases de datos que entrenan al vehículo IVVI 2.0

En la Figura 20 se muestra un diagrama de flujo que muestra la lógica que se sigue en el proceso del Etiquetado.

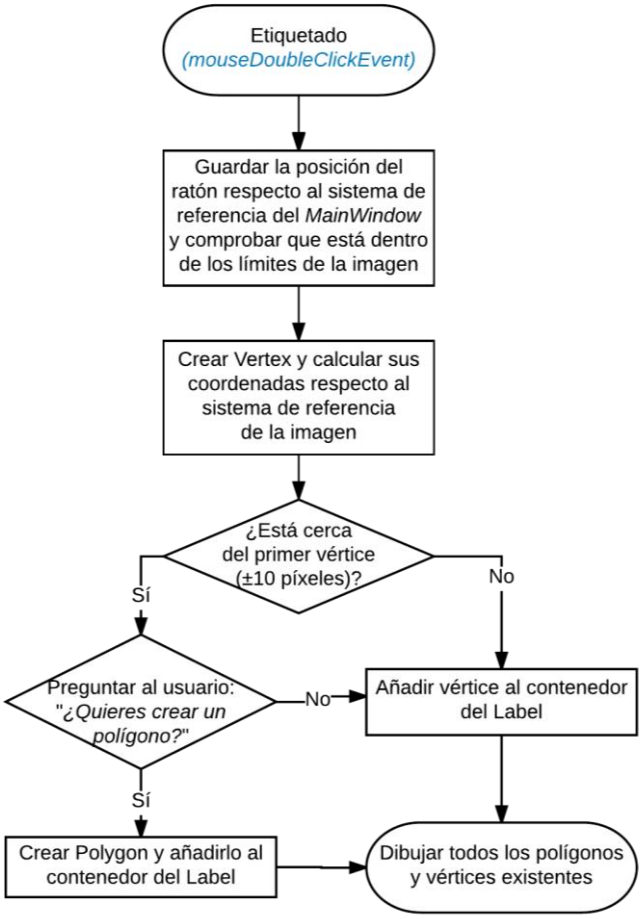

Figura 20 - Diagrama de flujo del proceso de Etiquetado

4.3.1 Cálculo de límites del etiquetado

Durante todo el proceso de etiquetado es importante tener definidas las dimensiones de la imagen o vídeo que se está etiquetando, así como su posición dentro de la herramienta. Es necesario calcular estos parámetros para poder comprobar que se está realizando las acciones de etiquetado dentro de la imagen, y no en otra área de la herramienta, así como para calcular la posición de los vértices respecto al sistema de referencia de la imagen, que se sitúa en la esquina superior izquierda de la misma.

En primer lugar, se obtendrán y guardarán las dimensiones de ancho (*width*) y alto (*height*) cada vez que se cargue una imagen o vídeo (se asume que todos los frames de un mismo vídeo tienen las mismas dimensiones, por lo que sólo se toman estos valores al abrir un vídeo), así como las del Área de visualización de la imagen o vídeo, denominado *labelImage.* Las imágenes y vídeos que son abiertos, siempre se cargarán centrados en este área. En caso de que la imagen o el vídeo tengan unas dimensiones superiores a las del *labelImage*, se redimensionarán manteniendo la proporción de la imagen al tamaño del área.

Se definen tres puntos en la imagen o frame, que facilitarán posteriormente el cálculo de las limitaciones: el centro de la imagen, el vértice superior izquierdo y el inferior derecho. Con estos dos últimos puntos, se obtiene el área de la imagen que delimita el área permitida para etiquetar.

Aunque el etiquetado se realiza sobre la imagen mostrada, las coordenadas de la posición del ratón al realizar doble click se obtienen en función del sistema de referencia de la ventana *MainWindow.* Debido a esto, es necesario realizar una conversión para conseguir la posición del ratón transformadas al sistema de referencia dela imagen. Para esta conversión es necesario tener en cuenta las dimensiones de la imagen, las del Área de visualización de la imagen o vídeo y los márgenes fijos que existen, como se puede observar en la Figura 21.

Figura 21 - Medidas y localización de las coordenadas limitantes de la imagen

Punto central

$$x_C = \frac{ancho_{imageLabel}}{2} + 13$$

$$y_C = \frac{alto_{imageLabel}}{2} + 27$$

Vértice superior izquierdo

$$x_{P1} = x_C - \frac{ancho_{imagen}}{2}$$

$$y_{P1} = y_C - \frac{alto_{imagen}}{2}$$

Vértice inferior derecho

$$x_{P2} = x_C + \frac{ancho_{imagen}}{2}$$

$$y_{P2} = y_C + \frac{alto_{imagen}}{2}$$

Ecuación 1 - Cálculo de las coordenadas limitantes de la imagen

El punto central no es necesario para el cálculo de las otras funciones, pero sí facilita los de las coordenadas limitantes.

Con estas coordenadas, se puede delimitar el área admitida para el etiquetado, quedando los intervalos definidos en la Ecuación 2.

$$x_{P1} \leq x_{labelling} \leq x_{P2}$$

$$y_{P1} \leq y_{labelling} \leq y_{P3}$$

Ecuación 2 - Intervalos de coordenadas admitidas en el etiquetado de imágenes

4.3.2 Obtención de la posición del ratón

Para la creación de un vértice, el usuario debe hacer doble click en el botón izquierdo de su ratón. Sin embargo, para que este vértice sea válido, deberá cumplir dos condiciones: tener el botón *Labelling* activado y que la doble pulsación esté dentro de los límites de la imagen explicados en el apartado anterior.

Para esto, es necesario saber dónde se encuentra el ratón en el momento en el que realiza la acción del doble click. En la librería de *Qt Creator*, dentro de la clase *QWidget*, existe una función llamada *mouseDoubleClickEvent* [31], que nos devuelve un Evento del ratón al hacer doble click con él. Gracias a este Evento, se puede obtener las coordenadas de la posición del ratón, que son dadas respecto al sistema de referencia del *MainWindow.*

Haciendo una simple comparación entre estas coordenadas y los puntos limitantes de la imagen, que fueron calculados respecto al mismo sistema de referencia, se puede comprobar si se cumple la condición de estar realizando la doble pulsación dentro de la imagen.

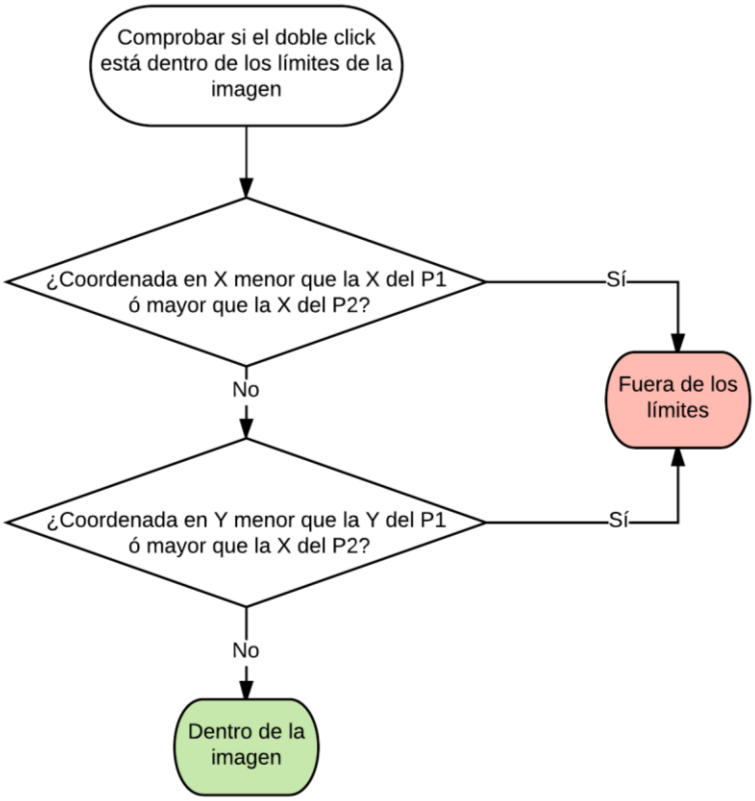

Figura 22 - Comprobación de las coordenadas del ratón respecto a los límites de la imagen

4.3.3 Creación de un vértice

Tras haber realizado la comprobación de que el ratón ha realizado la acción del doble click dentro de los límites de la imagen, se procede a crear el *Vertex.* Nos interesa que las coordenadas de este punto sean relativas a la imagen, y no a la ventana principal, por lo que será necesario realizar una transformación entre el sistema de referencia del *MainWindow* y el de la imagen, que se sitúa en la esquina superior izquierda (*P1)*.

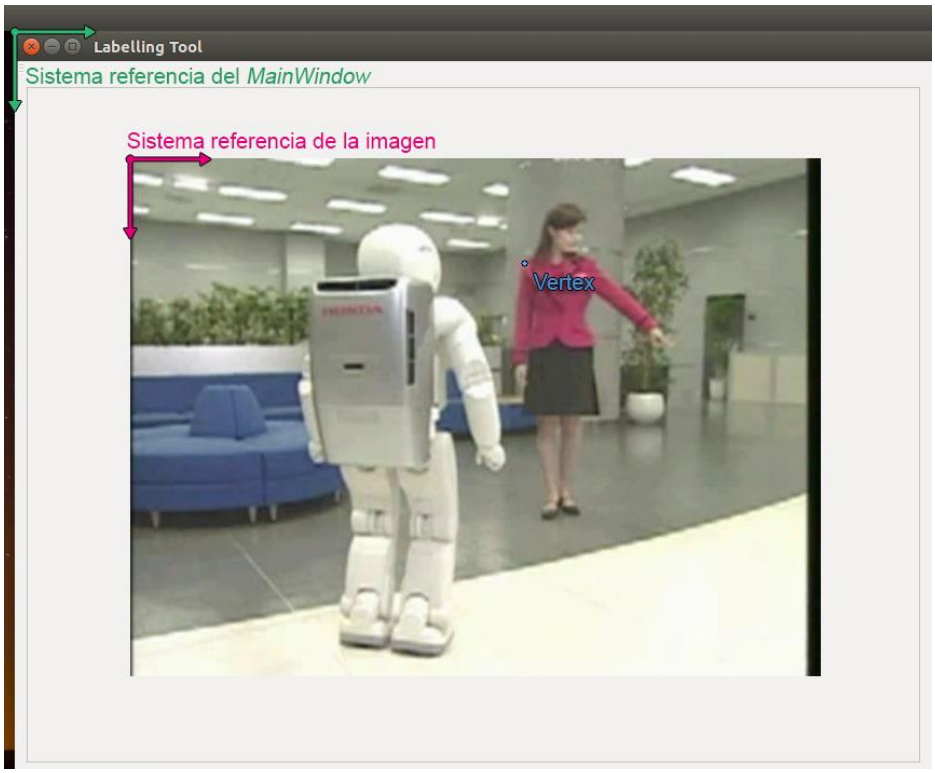

Figura 23 - Sistemas de referencia de la ventana principal y la imagen

En la Ecuación 3 se muestra el cálculo empleado para obtener las coordenadas del *Vertex*.

$$x_{CV} = x_{ratón} - x_{P1}$$

$$y_{CV} = y_{ratón} - y_{P1}$$

Ecuación 3 - Cálculo de las coordenadas del vértice

Siendo las coordenadas $x_{ratón}$ e $y_{ratón}$ el punto obtenido a partir de la posición del ratón, y x_{P1} e y_{P1} las coordenadas del sistema de referencia de la imagen, ambas posiciones respecto al sistema de referencia de *MainWindow.*

Este nuevo vértice es almacenado en el contenedor de *Vertex* de la etiqueta seleccionada, a la espera de que el usuario decida crear un nuevo polígono con los vértices que se encuentran en dicha etiqueta.

4.3.4 Creación de un polígono

Cuando el usuario desee crear un *Polygon*, y haya al menos tres vértices ya dibujados, solo deberá hacer doble click en el primer vértice para cerrar el polígono. Se permite un margen de ±10 píxeles alrededor del primer vértice, para facilitar la creación de los polígonos, ya que seleccionar exactamente la misma posición es complicado.

Al hacer doble click dentro de estos márgenes, se abre un diálogo auxiliar emergente, en el que se le pregunta al usuario si desea crear un nuevo polígono.

En caso afirmativo, se creará un nuevo objeto *Polygon* a partir de los vértices guardados en la etiqueta seleccionada que, al pasar a formar el polígono, son borrados del contenedor de vértices de la etiqueta, mientras que el *Polygon* se guarda en el contenedor de polígonos de la misma. Por el contrario, en caso negativo, se creará y guardará un nuevo *Vertex* en la posición seleccionada.

Figura 24 - Diálogo para la creación de un nuevo polígono

4.3.5 Dibujar vértices y polígonos

Cuando se crea un nuevo vértice o un nuevo polígono, cuando se elimina alguno de ellos o cuando se cambia una propiedad de alguna etiqueta, se llama a la función *paint().* Este método dibuja los polígonos de todas las etiquetas y los vértices de la seleccionada, sobre una copia de la imagen o frame original. De este modo se consigue solventar las siguientes situaciones:

- Al crear un nuevo *Polygon*, los vértices asociados a la etiqueta desaparecen de la imagen para ser reemplazados por el polígono dibujado.
- En las opciones de borrar el último vértice o el último polígono dibujado de la etiqueta seleccionada, sería necesario guardar las imágenes anteriores para poder volver a ellas en caso de que el usuario quisiera borrar algún elemento. Sin embargo, debido a que se dibuja todos los objetos existentes cada vez que se elimina uno, no es necesario guardar imágenes anteriores, ahorrando en memoria y simplificando el método de dibujar, además de ofrecer al usuario un método muy intuitivo para corregir errores.
- Cuando se edita los atributos de una etiqueta, sobre todo si se modifican el color o la propiedad del *BoundingBox*, el dibujo de los polígonos asociados a dicha etiqueta necesita ser actualizado, para mostrar correctamente los valores del *Label*. Gracias a que los polígonos y los vértices son dibujados al completo, estos cambios se muestran sin mucha complicación de código, además de evitar confusiones al usuario, que comprueba de forma visual que los cambios se han realizado correctamente.
- De manera similar, si una etiqueta es eliminada, los polígonos y vértices pertenecientes a la misma desaparecen, por lo que también deberán borrarse de la imagen dibujada. Esto sería complicado de conseguir si no se pintasen todos los elementos constantemente, ya que se deberían realizar capas superpuestas por cada polígono y por cada vértice, que serían eliminadas una a una las que estuvieran asociadas a la etiqueta borrada. Todo ello supondría una almacenamiento de memoria superior, además de una mayor complejidad de programación en el código.

Para dibujar los polígonos rellenos del color de la etiqueta asociada, se hace uso de la función *fillPoly* [32] de la librería OpenCV, en la que se envían con parámetros la copia de la imagen original, un puntero a los vértices que

componen el polígono, el número de vértices, se le indica que hay un único contorno y el color de la etiqueta.

Si la etiqueta a la que está asociada el polígono dibujado tiene la propiedad *BoundingBox* activada, se utiliza la función *boundingRect* [33] de la librería OpenCV, a la cual enviamos el contenedor con todos los vértices que componen el polígono y nos devuelve un objeto *Rect* [34], es decir, el mínimo rectángulo delimitador del polígono. Para dibujarlo en la imagen, se hace uso de la función *rectangle* [32], también de OpenCV, en la que se envía la copia de la imagen original, el rectángulo calculado, el color negro para diferenciarlo del polígono y el grosor de la línea.

Los vértices se representan con círculos rellenos del color de la etiqueta, con un borde exterior negro para distinguirlo de la imagen. Para ello, se hace uso de la función *circle* [32] de la librería OpenCV, la cual recibe como parámetros la imagen, las coordenadas del vértice, el radio del círculo y el color de la etiqueta.

Para marcar los lados de los polígonos, para facilitar la visión para el usuario de qué forma va tomando el polígono mientras etiqueta, se dibujan líneas entre un vértice y el inmediatamente anterior a él en el contenedor de vértices de la etiqueta. Esta línea se dibuja haciendo uso de la función *line* [32], igualmente de la librería OpenCV, que recibe la copia de la imagen original, las coordenadas del vértice y las del anterior, el color del *Label* y el grosor deseado.

4.4. ALMACENAMIENTO DE LA INFORMACIÓN

En este apartado se van a explicar los diferentes formatos disponibles para exportar la información utilizados por la herramienta, que constituirán las bases de datos anotadas para el entrenamiento de algoritmos de percepción, así como los ficheros de almacenamiento empleados para guardar la información de los elementos empleados para el etiquetado en la herramienta, desarrollando la estructura de los archivos XML y el método empleado para guardar y cargar estos documentos.

4.4.1 Dataset

Cuando el usuario desee exportar los datos que ha creado durante el etiquetado, debe ir a la pestaña "*Export*" y seleccionar la opción *Dataset*. Como se puede ver en la Figura 12, la herramienta abre un diálogo donde el usuario puede seleccionar las opciones que más le interesen para crear el *Dataset* de la imagen etiquetada en ese momento.

Las opciones que se ofrecen al usuario para crear las distintas salidas siguen los mismos formatos de las bases de datos más empleadas en el Laboratorio de Sistemas Inteligentes en el campo de investigación de la visión por computador, es decir, *Cityscapes* y *KITTI* (ver apartados 2.3.3 y 2.3.4, respectivamente).

Para el almacenamiento de las salidas en las que se crea una imagen a partir de la propiedad solicitada de los polígonos, se hace uso de la clase *Mat*[35], de la librería de *OpenCV*. Esta clase es una de las más utilizadas de esta librería dentro de la herramienta, ya que ésta permite con relativa sencillez de código realizar sobre ella las transformaciones necesarias para mostrar el avance del etiquetado, como por ejemplo los círculos, líneas o polígonos que se dibujan sobre las imágenes mostradas a partir de este objeto.

- ***Colors***:

  Se crea un objeto *Mat* que representa una imagen de fondo negro, donde se dibujan los polígonos de todas las etiquetas, pintados con el color de la etiqueta a la que está asociado cada uno de ellos. La salida se guarda en la misma carpeta donde se encuentra la imagen o vídeo original, añadiendo la terminación "_colors.jpg" al nombre original.

- ***IDs***:

  Se dibujan todos los polígonos existentes con el nivel de gris de la etiqueta a la que están asociados, en una nueva imagen (objeto *Mat*) de fondo negro. Ésta es guardada en la dirección de la imagen original, y con la terminación "_IDs.jpg" en el nombre.

- ***Instanciables***:

De forma similar a la salida anterior, en este caso únicamente se dibujan los polígonos que pertenezcan a las etiquetas que tengan activada la propiedad de *Instanciable,* dibujados con su nivel de gris sobre un fondo negro. Esta imagen se guarda en la misma dirección del archivo original, esta vez con la terminación "_instanciables.jpg" en el nombre.

- ***XML Polygones***:

Esta salida genera un documento XML que guarda toda la información relacionada con los polígonos: las propiedades de las etiquetas, los polígonos que están asociados a cada una de ellas y los vértices que los componen. Esta forma de almacenaje que explicará con mayor detalle en el apartado 4.4.2 Documentos XML.

- ***Current Frame***:

En caso de estar etiquetando el frame de un vídeo, en el diálogo de la generación del *Dataset* se activa esta opción (Figura 12), que genera una imagen a partir de ese frame y la guarda en la dirección donde se encontraba el vídeo original, añadiendo el número del frame en el que se encuentra. Esta salida es útil si el usuario desea acudir a ciertos frames en concreto de un vídeo, lo que evita la necesidad de cargar el vídeo entero y tener que navegar por él buscando los frames exactos. El usuario podrá guardar los fotogramas deseados y después abrir los que le interesen en forma de imagen.

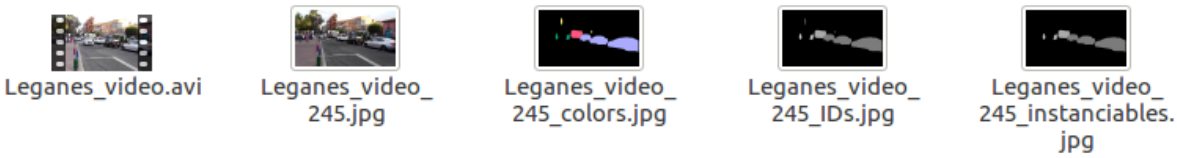

Figura 25 - Dataset guardado en la dirección del archivo original

De izquierda a derecha, vídeo original y las salidas de Dataset: current frame, colors, IDs e instanciables.

4.4.2 Documentos XML

La herramienta emplea este tipo de meta-lenguaje para guardar la información relativa a las etiquetas existentes, con todas sus propiedad, los polígonos creados por el usuario y los vértices que componen cada uno de ellos.

En este proyecto, la lectura y escritura de los archivos XML se realiza mediante la librería TinyXML-2, que permite crear documentos de forma relativamente sencilla y eficiente, consiguiendo almacenar la información necesaria en un reducido espacio de memoria.

Esta forma de guardar la información se usa en el software en dos tipos de documentos: uno en el que se guarda la información de los *Labels* que se encuentran en el Panel de Etiquetas, y otro en el que se almacena la información relativa a los polígonos que creados en el etiquetado.

- **Labels:**

  Como ya se ha explicado en apartados anteriores, en la herramienta de etiquetado están implementadas las opciones de cargar y guardar las etiquetas, como *Open Labels* and *Export Labels*, respectivamente. Con esto se pretende dar al usuario la opción de reutilizar etiquetas que han sido creadas con anterioridad, que son utilizadas con frecuencia o quieren guardarse para continuar etiquetando, por ejemplo, distintos frames de un vídeo en otra ocasión.

  Estas opciones son muy útiles, ya que el usuario ahorrará tiempo al no tener que crear siempre las mismas etiquetas. También puede cargar etiquetas que tenga de usos anteriores, y editarlas a la necesidad de ese momento, dando mayor dinamismo y eficiencia al proceso de etiquetado.

  Estos documentos son escritos en el lenguaje XML, haciendo uso de la librería *TinyXML-2*. Se debe seguir un orden secuencial tanto en el proceso de lectura como en el de escritura, para evitar errores en los atributos de la

etiqueta. Por ejemplo, si al almacenar los *Labels* primero escribimos el nombre del mismo y después su ID, al leer este documento deberemos seguir ese mismo orden.

Cuando el usuario decida guardar los *Labels* presentes en el Panel de Etiquetas se abre un diálogo que le permite elegir la ubicación y el nombre que le quiere dar al archivo, al que se le asigna la extensión "*.xml*". Esto facilita que el usuario pueda reconocer qué grupo de etiquetas hay en cada documento, según sus propios códigos de organización.

Al seleccionar la opción *Open Labels*, el diálogo le permite navegar hasta encontrar la dirección del archivo, mostrando únicamente los de extensión "*.xml"* para simplificar la búsqueda al usuario. Estas etiquetas se añaden por debajo al contenedor de *Labels* de la página principal, independientemente de si ya hay o no etiquetas cargadas.

La estructura que siguen los documentos en los que se almacenan etiquetas es la siguiente:

En el documento, lo primero que se almacena es el número de etiquetas que hay en el contenedor del *MainWindow*. Después se abre la primera etiqueta y se inserta el nombre y su número ID. Al llegar al color de la etiqueta, se desglosa en los tres niveles del RGB: el del rojo, el del verde y el del azul. A continuación, se almacena si están desactivadas o activadas, en código binario de 0 y 1, respectivamente, las propiedades de *Instanciable* y *BoundingBox.* Al finalizar esto, se abre la siguiente etiqueta, y se repite el proceso para todas las etiquetas existentes.

- **Polygones:**

Una de las salidas implementadas en la opción de *Export Dataset* de la herramienta, es la posibilidad de crear un documento XML que almacene todos los polígonos que están etiquetados (ver Figura 12). Con esto se ofrece al usuario la posibilidad de recuperar imágenes etiquetadas anteriormente,

lo que le permite poder realizar este trabajo en diferentes momentos. La utilidad de esta opción es significativa, ya que ahorra tiempo al usuario al devolverle al punto en el que dejó el proceso de etiquetado anteriormente.

Este documento se escribe en lenguaje XML, haciendo uso de la librería *TinyXML-2*. De nuevo, el orden secuencial en el que se escriben las propiedades en el documento debe ser el mismo para leerlo, con la función de evitar errores en la carga de los elementos que se crean a partir del documento.

Cuando el usuario decida guardar los *Polygones* presentes en la imagen etiquetada, se crea un documento XML con la dirección de la imagen original, a la que se le añada al final el nombre "_Polygones" y se le asigna la extensión "*.xml*". Si el usuario está etiquetando un vídeo, a la dirección original se le añade el número del frame en el que se encuentra, quedando de la siguiente manera:

"[dirección original]_[número del frame]_Polygones .xml"

Esta nomenclatura se utiliza para facilitar al software la búsqueda de este documento cuando se abra de nuevo la imagen o se llegue a ese frame del vídeo. El usuario no tiene que buscar el archivo de los polígonos cuando quiera cargarlo, si no que el programa lo hará por él. Cuando se abre una imagen o se pasa a un frame del vídeo, el programa busca en la carpeta donde se encuentra dicho elemento en búsqueda de uno que tenga la nomenclatura de los documentos de *Polygones*. En caso de encontrarlo, la herramienta pregunta al usuario si desea cargar estos polígonos.

Si el usuario acepta, se comprueba si ya existen etiquetas en el Panel de Etiquetas. Si las hubiera, se da al usuario la posibilidad de sustituir estas etiquetas por las que vienen guardadas en el documento de *Polygones*, o si simplemente quiere añadirlas por debajo de la lista a las ya existentes. Esto se hace debido a la posibilidad de que las etiquetas nuevas sean las mismas

que las que ya están cargadas, o tengan propiedades similares. Se deja a juicio del usuario decidir si eliminar las antiguas o añadir las nuevas, con la posibilidad de tener que editarlas para que no se confundan entre ellas.

Independientemente de esta última decisión, posteriormente el software carga las etiquetas guardadas en el documento XML al contenedor del *MainWindow*, y crea los polígonos asociados a cada una de ellas a partir de los valores de los vértices.

En el documento, la estructura para almacenar los atributos de las *Labels* es el mismo que el visto en el apartado anterior. Después de almacenar si la propiedad de *BoundingBox* está activada, se guarda el número de polígonos que tiene asociada esa etiqueta.

Se abre el primer polígono, y se almacena el número de vértices que lo compone. Se abre el primer vértice y se guardan las coordenadas en X y en Y del mismo. Tras esto, se repite este proceso hasta almacenar todos los vértices que componen el polígono.

Si la etiqueta a la que está asociado el polígono tiene activada la propiedad del *BoundingBox,* se guardarán las dimensiones del rectángulo que abarca el polígono que estamos almacenando: las coordenadas de su esquina izquierda superior, su ancho y su alto.

Al finalizar esto, se abre el siguiente polígono, y se repite el esquema para todos los polígonos asociados a la etiqueta. Posteriormente, se irán abriendo y almacenando todas las etiquetas existentes siguiendo este patrón.

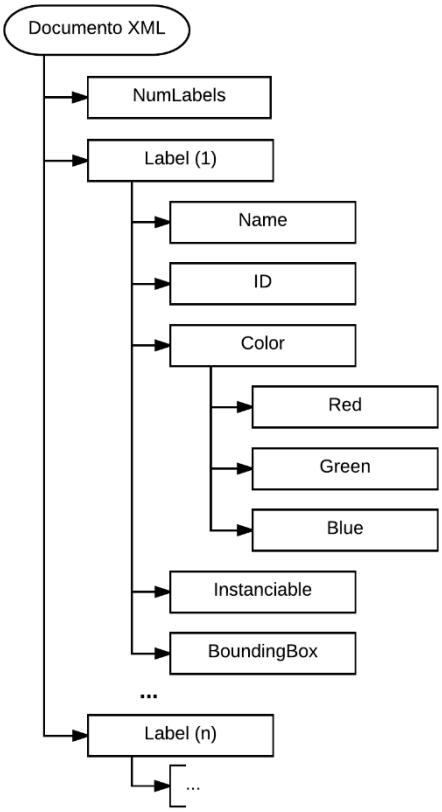

Figura 26 - Estructura del documento XML para Labels

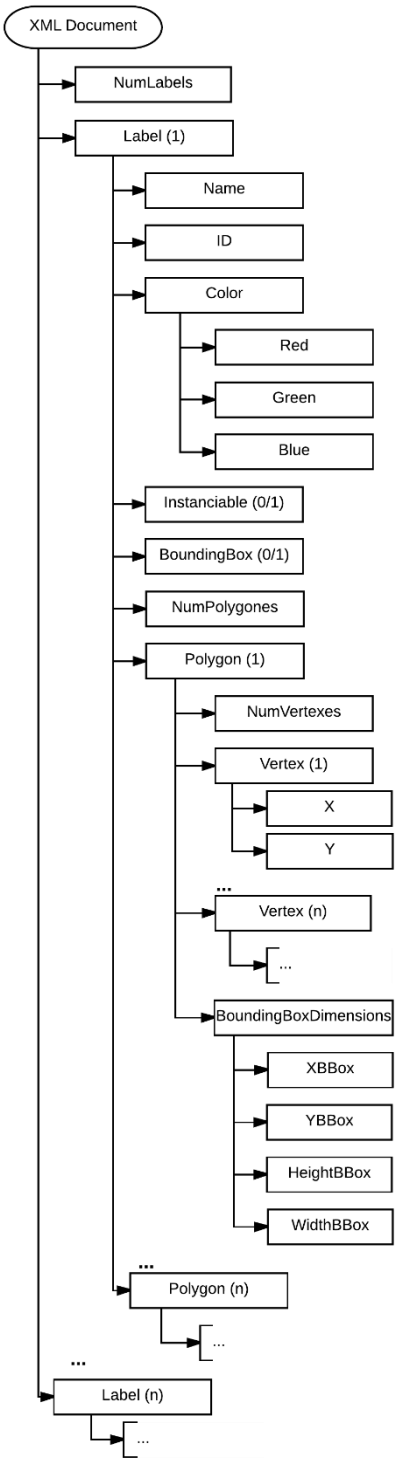

Figura 27 - Estructura del documento XML para Polygones

5. RESULTADOS

En este capítulo se va a mostrar un ejemplo del uso de la aplicación de la herramienta de etiquetado. Se utiliza un vídeo realizado en las inmediaciones del campus de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, en Leganés. Se anotan algunos ejemplos de distintas categorías: dos señales de tráfico aplicables a ese tramo de carretera, cuatro vehículos, haciendo diferencia entre turismos y autobuses, y los peatones que están cruzando o los que se encuentran cerca y potencialmente podrían cruzar. Se podrían hacer tantas etiquetas como se desease, llegando incluso a etiquetar todos los píxeles de la imagen, como ocurre en las imágenes de la base de datos *Cityscapes* [20], pero se muestran únicamente unos pocos polígonos para una mejor comprensión y visualización de la obtención de los resultados.

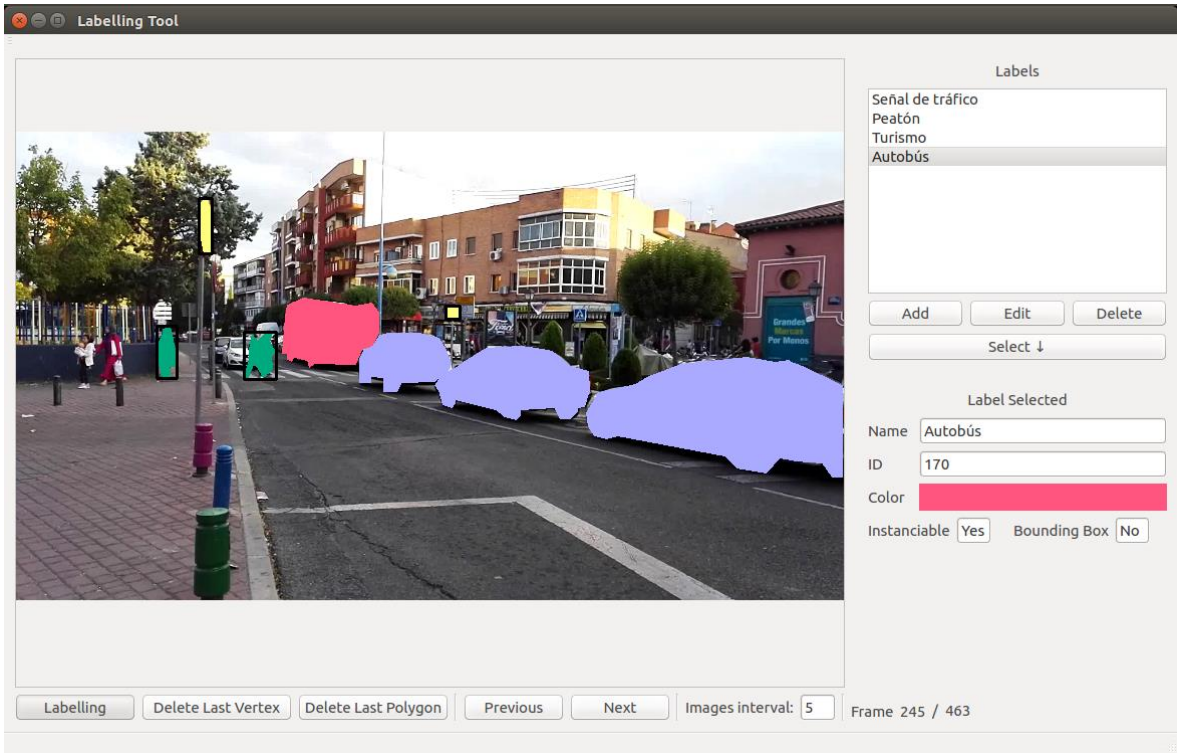

Figura 28 - Ventana principal de la herramienta con frame de vídeo etiquetado

En este caso, se han solicitado exportar las etiquetas que han sido creadas desde cero cuando el usuario ha abierto la herramienta. Éstas etiquetas son almacenadas bajo el

nombre "*Labels Leganés.xml*". Además, se han seleccionado todas las salidas ofrecidas por el programa para exportar el *Dataset* de este frame.

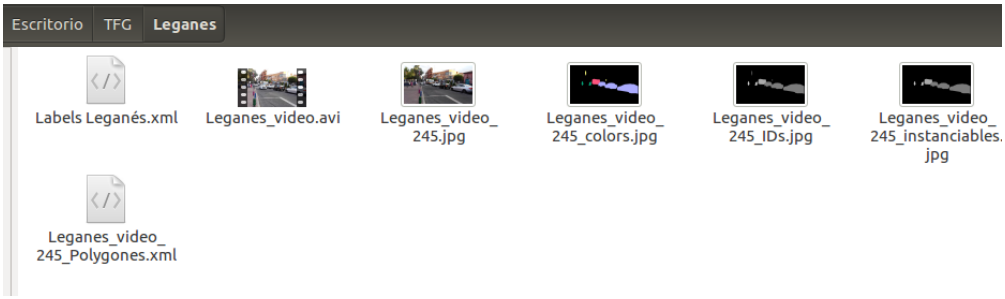

Figura 29 - Dataset exportado automáticamente a la ubicación original del vídeo

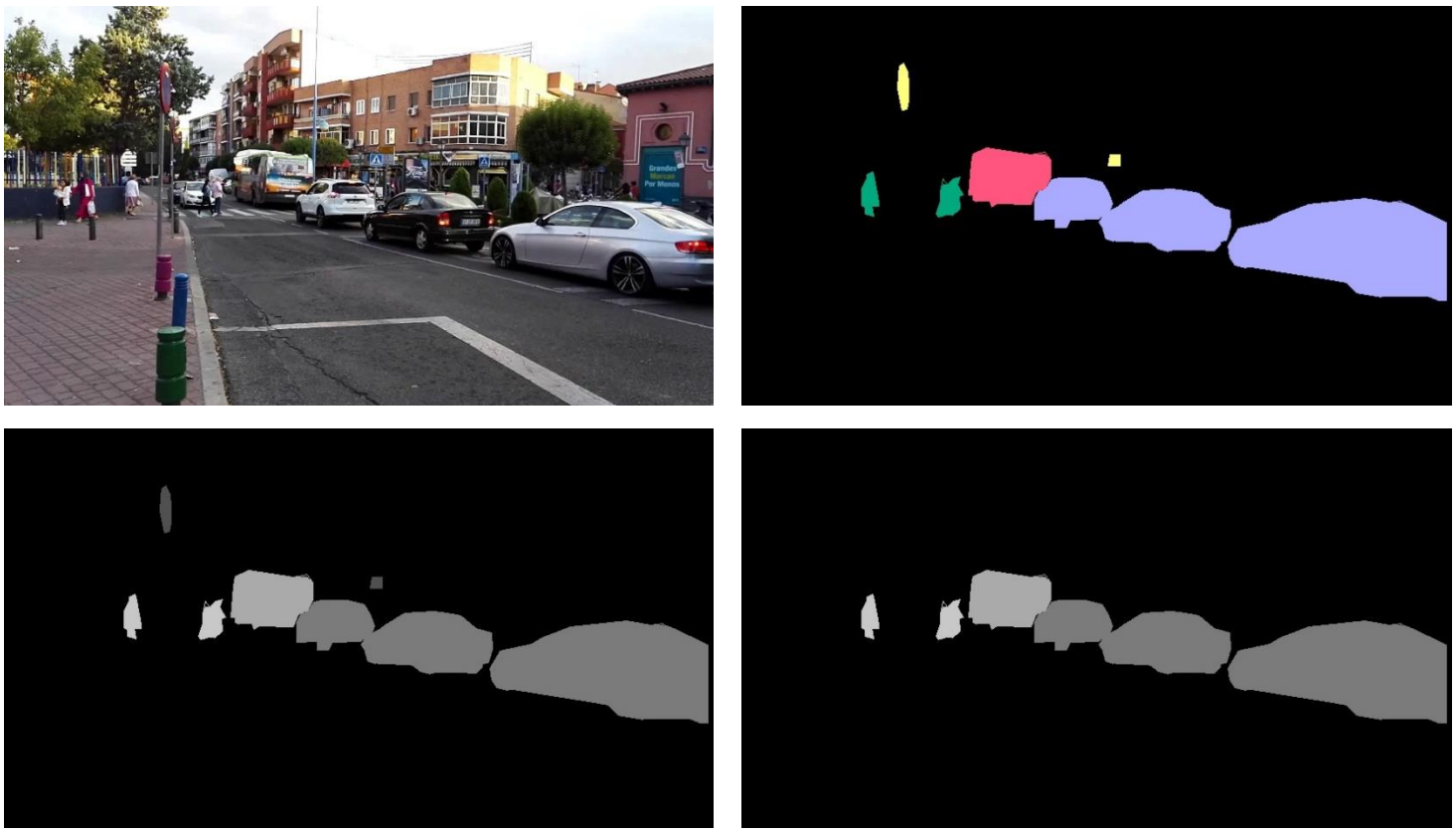

Figura 30 - Imágenes exportadas que conforman el Dataset

Como se puede ver en la Figura 29 la herramienta crea las salidas solicitadas para la formación del Dataset, y son exportadas automáticamente a la misma dirección donde se encuentra el vídeo original. En la Figura 30, se muestra cómo la herramienta hace una copia del fotograma en el que se ha estado etiquetando (arriba a la izquierda), además de crear tres imágenes:

- Imagen de los polígonos con los colores a los que están asociados. Arriba a la derecha se puede ver, por ejemplo, que los tres coches tienen el mismo color, mientras que el autobús está dibujado en otro, ya que el usuario ha creado dos categorías diferentes.

- Imagen de los polígonos con el nivel de gris que le corresponde. De nuevo, se puede ver, abajo a la izquierda, cómo, por ejemplo, los peatones están coloreados con un mismo nivel.

- Imagen de los polígonos instanciables con el nivel de gris que le corresponde, abajo a la derecha. Podemos ver cómo las señales de tráfico etiquetadas desaparecen de esta salida, ya que esa etiqueta no tenía la propiedad activada.

Figura 31 - Estructura de los archivos XML que exportan las etiquetas (izquierda) y los polígonos (derecha)

En la Figura 31 se muestra el comienzo de los archivos que se generan al exportar las etiquetas que se encuentran en el Panel de etiquetas, o al seleccionar la opción de la formación del *Dataset* de exportar la posición de los polígonos.

6. MARCO REGULADOR

En el momento en el que se redacta el presente documento, la legislación española no recoge ni regula el uso de vehículos completamente autónomos en las carreteras nacionales, aunque sí a los que integran un sistema semiautomático como el integrado en los coches Tesla. Tan solo se permiten, desde noviembre de 2015, las pruebas de los vehículos autónomos con una autorización previamente solicitada a la DGT, en la que la norma establece quiénes podrán realizar dichas pruebas:

"Según la instrucción, vehículo autónomo es todo aquel que dispone de capacidad motriz equipado con tecnología que permita su manejo o conducción sin precisar la forma activa de control o supervisión de un conductor, tanto si dicha tecnología autónoma estuviera activada o desactivada de forma temporal o permanente.

Podrán solicitar la autorización para la realización de pruebas y ensayos los fabricantes de vehículos autónomos, sus carroceros y los laboratorios oficiales, así como los fabricantes o instaladores de la tecnología que permita al vehículo plena autonomía, las universidades y consorcios que participen en proyectos de investigación." [36]

El rápido avance de estas tecnologías apremia a los legisladores, que necesitan redactar un marco legal que regule este tipo de vehículos, algo que no es tarea fácil. Este código vial deberá adaptarse a las características especiales que rigen la conducción autónoma, y a la vez conseguir que sean lo suficientemente flexibles como para poder adaptarse a los cambios que se produzcan en esta tecnología, ya que, probablemente, ocurran más rápido de lo que las leyes acostumbran a cambiar [37].

Además del avance legislativo en materia de regulación del tráfico, otro aspecto a considerar es la necesidad de modificar las leyes que regulan el seguro obligatorio para conducir, para que también se contemplen este tipo de vehículos. Para esto, es necesario marcar las responsabilidades de la conducción en caso de accidente. En este campo, a finales de agosto de 2017 se aprobó en Alemania "el primer código ético del mundo para vehículos autónomos" [38].

7. ENTORNO SOCIO-ECONÓMICO

7.1. IMPACTO SOCIO-ECONÓMICO

El objetivo de conseguir buenas y amplias bases de datos va a permitir entrenar correctamente los sistemas de visión artificial de los vehículos autónomos, que se estima que sean una revolución para la sociedad en un futuro no tan lejano.

El principal impacto que tendrá la conducción autónoma, y uno de los motivos más importantes por lo que se está invirtiendo en el desarrollo de esta tecnología, es el gran impacto en la seguridad vial que supondrá utilizar este tipo de vehículos. Se estima que, actualmente, el 93% de los accidentes de tráfico son causados por errores humanos[39], por lo que, si la conducción fuera controlada por un buen sistema de navegación de una máquina, ese factor quedaría totalmente reducido, salvando millones de vidas. Actualmente, según las estadísticas, se produce un accidente cada cien mil kilómetros, y 1,2 millones de personas en todo el mundo pierden la vida cada año. Se estima que con la conducción autónoma la cifra se reduciría a un accidente cada diez millones de kilómetros[40]. Además, la tecnología de un vehículo autónomo está conectada con los sistemas de los de su entorno, compartiéndose información de, por ejemplo, qué trayectoria están tomando. Esto ayuda a prever los movimientos de los vehículos cercanos, por lo que se evitarán colisiones accidentales por cambios de carril, cruces, rotondas, etc.

Este tipo de vehículos utiliza un mejor rendimiento del motor y, junto al hecho de que está pensado que sean híbridos o eléctricos, la cantidad de combustible fósil que se utilizará se reducirá drásticamente, estimándose un 56% menos de emisiones de combustible al medio ambiente[39]. Además, el espacio necesario para aparcar los vehículos también se verá disminuido, ya que los vehículos se aparcarán solos, pudiendo ajustar más el espacio entre ellos. Este espacio podría invertirse en espacios verdes que, añadiéndose a la disminución de la utilización de carburantes para los vehículos y el mayor rendimiento en el uso del motor, supondrían una significativa mejora en la calidad del aire y la reducción de la

contaminación. Además, los atascos serán casi inexistentes debido a la reducción de accidentes, una mejor conducción y una mayor eficiencia en el uso de las carreteras. Por todo ello, es evidente el enorme impacto positivo que tendría en materia medioambiental la utilización de este tipo de vehículos.

Los vehículos autónomos permitirán que personas de avanzada edad, los que sufren de movilidad reducida o tienen algún tipo de discapacidad, puedan conseguir una mayor independencia al no necesitar de otra persona para poder desplazarse. Este punto es especialmente interesante, ya que se está tendiendo a una sociedad cada vez más envejecida, donde se estima que, en unas décadas, el número de personas mayores de 80 años se triplicará [39].

Estos vehículos están diseñados para interpretar las señales de tráfico del entorno y respetarlas, por lo que las multas de tráfico se reducirán [39]. Esto ahorrará costes tanto para el estado en la tramitación que suponen estas sanciones, como para los propios ciudadanos. Tampoco serán necesarios los controles, ya que no supondrá un peligro para la seguridad vial que una persona se encuentre en estado de embriaguez dentro del vehículo, pues no será él quien esté conduciendo ni tomando las decisiones.

Las personas podrán trabajar de camino a casa, en vez de invertir ese tiempo en la conducción, lo que aumentará las horas de productividad de una persona. Se estima que actualmente personas en todo el mundo pierden 5,5 mil millones de horas potencialmente productivas por el tiempo que invierten en desplazarse, muchas veces agravado por el tráfico y los atascos [39]. Esto podría traducirse en una descentralización de la sociedad, ya que las personas podrán optar por vivir más lejos del trabajo sin que ello suponga perder unas horas que podrían invertir en su productividad [40]. Esto acarrearía una menor contaminación, además de un mejor uso del espacio disponible y una menor concentración de vehículos en las autovías que actualmente tienden a congestionarse. Además, la opción de hacer parte del trabajo en el trayecto a casa, podría dar la opción de mayor flexibilidad en los horarios laborables, pudiéndose adaptar a vida de ocio y familiar de cada persona,

algo que actualmente supone un problema tanto para los trabajadores como para las compañías [41].

Los seguros de vehículos deberán adaptarse a la nueva situación, bajando sus precios ante la reducción masiva de los accidentes de tráfico. Lo mismo ocurrirá con el precio de la vivienda, que tenderá a unificar las ofertas de precio en todas las zonas debido a la menor concentración de personas en las áreas más céntricas de las ciudades [40].

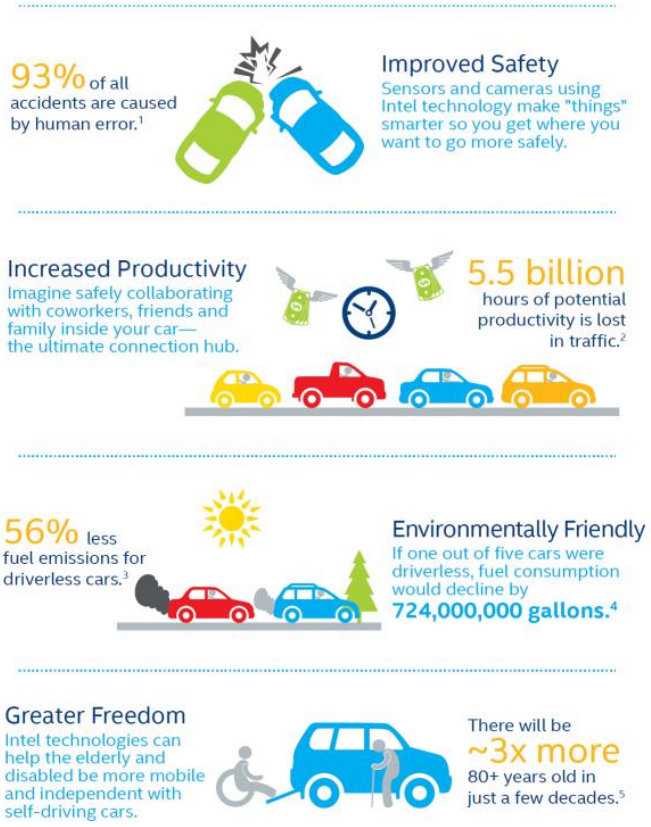

Figura 32 - Impactos de la conducción autónoma en la sociedad

En una entrevista realizada al CEO de Mercedes Benz [40], comenta su visión del futuro de la sociedad debido al gran avance de la tecnología, desde la casi desaparición de algunos puestos de trabajos tradicionalmente importantes, como el de un abogado, hasta cómo las impresoras 3D transformarán la industria agrícola. En cuanto a la conducción autónoma, trata en gran medida los puntos expuestos en este apartado, aunque va más allá: asegura que habrá hasta un 95% menos de vehículos, ya que la gente tenderá a utilizar vehículos compartidos o compañías del

estilo de Uber, donde un vehículo autónomo te recogerá donde estés y te llevará a donde solicites. Esto supondría unos resultados más drásticos de los expuestos en este apartado, ya que habría mucho más espacio disponible, menor tráfico y los accidentes automovilísticos serían casi inexistentes.

Aunque la tecnología actual parece sí poder llegar a este futuro que se ha presentado y, además, hacerlo en relativamente pocos años, la verdadera cuestión es si la sociedad actual, que cada vez tiende más al individualismo y al materialismo, y donde en muchas ocasiones existe un recelo hacia la robótica y las tecnologías autónomas por la influencia de los futuros apocalípticos mostrados en películas y novelas, está realmente dispuesta a aceptar todos estos cambiamos y adaptarse a las nuevas condiciones.

7.2. PRESUPUESTO

En este capítulo se va a calcular el presupuesto estimado para el desarrollo del presente proyecto, teniendo en cuenta el software y el hardware empleados, así como los costes en recursos humanos de los ingenieros implicados. No se valoran las obligaciones fiscales de estas personas ni las posibles retenciones aplicadas al personal.

Costes de Material asociados al desarrollo del proyecto y la redacción de la memoria								
Código	Unidad	Descripción	Medición	Amortización (meses)	Meses utilizado	Precio unitario con IVA (€)	Coste total para proyecto (€)	IVA para proyecto(€)
Capítulo 1: Hardware utilizado								
1.1	UD	Ordenador portátil ASUS F556U	1	48	6	650,00 €	81,25 €	17,06 €
						Total capítulo 1	81,25 €	17,06 €
Capítulo 2: Software empleado en el proyecto								
2.1	UD	Qt Creator	1	N.A.	6	0,00 €	0,00 €	0,00 €
2.2	UD	OpenCV	1	N.A.	6	0,00 €	0,00 €	0,00 €
2.3	UD	TinyXML-2	1	N.A.	3	0,00 €	0,00 €	0,00 €
2.4	UD	Linux Ubuntu 16.04	1	N.A.	6	0,00 €	0,00 €	0,00 €
2.5	UD	Servidor Bitbucket	1	N.A.	6	0,00 €	0,00 €	0,00 €
						Total capítulo 2	0,00 €	0,00 €
Capítulo 3: Software empleado en la memoria								
3.1	UD	Microsoft Office 2016 (versión académica)	1	N.A.	1	0,00 €	0,00 €	0,00 €
						Total capítulo 3	0,00 €	0,00 €
						TOTAL COSTES MATERIAL	81,25 €	17,06 €

Costes de Personal					
Nombre	Categoría	Horas trabajadas	€/hora	Coste final	IVA
Inés M. Carpio Coll	Ingeniero Junior	600	15,00 €	9.000,00 €	-
Jorge Beltrán de la Cita	Ingeniero Senior	40	40,00 €	1.600,00 €	-
			TOTAL COSTES PERSONAL	10.600,00 €	-

TOTAL PRESUPUESTO DEL PROYECTO	10.681,25€	17,06€

Figura 33 - Presupuesto total del proyecto

8. TRABAJOS FUTUROS

En este capítulo se van a explicar algunas líneas futuras que podrían tomarse en el desarrollo de esta herramienta. Estas propuestas ayudarían a simplificar más aún el proceso del etiquetado, lo que añadiría más valor de la herramienta.

8.1. EXTRAPOLACIÓN DE OBJETOS ENTRE IMÁGENES

Una ampliación muy interesante para el estado actual de la herramienta sería dar la posibilidad de extrapolar los polígonos creados por el usuario a otros frames del vídeo, o a las siguientes imágenes de la secuencia que se ha cargado. De esta manera, el usuario situaría la posición de un polígono ya creado en el nuevo frame y la herramienta podría calcular la posición de dicho objeto en los frames que hayan quedado en medio.

Esto ayudaría a agilizar enormemente el proceso del etiquetado, aunque también supondría un alto riesgo de imágenes erróneamente etiquetadas si se realizase entre frames distantes en el tiempo, pues no sería seguro que el objeto siguiera una trayectoria lineal.

Para facilitar el proceso de extrapolación, lo más sencillo sería utilizar las medidas de las BoundingBoxes asociadas a los polígonos, por lo que los objetos que quieran ser extrapolados deberán tener activada esta propiedad en su etiqueta.

8.2. FORMAS PREDEFINIDAS DE POLÍGONOS

Otra forma de agilizar y simplificar el proceso de etiquetado sería ofrecer al usuario una gama de posibles formas de polígonos. De la misma manera que tiene implementada esta opción la herramienta VGG (apartado 2.4.6), la posibilidad de tener predefinidas formas como círculos, elipses o rectángulos, pero sin perder la opción del polígono de forma libre, simplificaría el etiquetado de objetos geométricos repetitivos, como pueden ser las señales de tráfico.

9. CONCLUSIONES

En este capítulo se van a explicar a las conclusiones a las que se ha llegado tras la finalización de este proyecto, así como una valoración personal de los conocimientos obtenidos

El objetivo de este Trabajo de Fin de Grado es el desarrollo de una herramienta para el etiquetado denso de imágenes, que permita obtener los formatos de imágenes empleados en las bases de datos más utilizadas para el entrenamiento de las plataformas de investigación de la conducción autónoma del LSI de la UC3M.

Se ha desarrollado una herramienta funcional, en la que se ha priorizado la simplificación de todo el proceso de etiquetado, tratar de conseguir que esta tarea resulte más sencilla y ágil de realizar para los investigadores del Laboratorio.

Para ello se ha diseñado una interfaz gráfica sencilla e intuitiva, que además notifica de posibles errores y pregunta al usuario antes de realizar un cambio grande en los datos del etiquetado. Todo ello está implementado con la idea de evitar que un investigador pueda perder todo el trabajo empleado en el etiquetado de una imagen por pulsar por error un botón, por ejemplo.

También se ha buscado que las anotaciones se ajusten al nivel de precisión buscado por el usuario en cada momento, tanto si se desea realizar un etiquetado preciso que recoja la forma exacta de los objetos, o se considera que puede hacer formas más simples para conseguir realizar el proceso en menor tiempo. El alto grado de personalización de las etiquetas es otro aspecto positivo de esta etiqueta, ya que permite generalizar en una familia de objetos o focalizar en los distintos tipos, como por ejemplo, el caso de etiquetar como vehículo, o distinguiendo si se trata de un turismo, un camión, un autobús, una bicicleta… Incluso, podrían diferenciarse entre un vehículo circulando y otro aparcado, con distintas etiquetas.

Con todo ello, considero que se han logrado alcanzar los objetivos de este proyecto, pues se han implementado todas las necesidades y las soluciones a los problemas propuestos por el Laboratorio en la asignación del TFG, además de ampliar en aspectos

visuales de la aplicación o de simplificación del etiquetado, como por ejemplo las posibilidades de borrar vértices y polígonos en caso de no estar posicionados correctamente.

Anteriormente había trabajado con el software de *Qt Creator* durante el Grado, pero nunca enfocándose al aspecto del desarrollo de una interfaz gráfica, sino como mero compilador de código. Los conocimientos que he obtenido gracias al proyecto en este aspecto son amplios, dándome la oportunidad de desarrollar otras herramientas en el futuro. Este es en el sector donde más he adquirido nuevos conocimientos.

Además, nunca había utilizado el lenguaje de XML para el almacenamiento de información. He podido obtener unos conocimientos básicos que me han permitido ver el gran potencial de este tipo de documentos, que simplifican mucho el proceso de guardar la información necesaria, respecto a otras maneras aprendidas durante el Grado universitario, que eran bastante complicadas.

También he ampliado mi conocimiento sobre el procesado de imágenes y las bases de datos, además de plantearme la tecnología de la conducción autónoma desde un punto de vista más cercano a mis conocimientos, cosa que antes de este proyecto consideraba este tema demasiado complicado para intentar formar parte de algo relacionado con ello.

Adicionalmente, he conseguido coger mucha soltura programando en C++. Este lenguaje de programación me resultó complicado cuando lo estudié en un principio, pero actualmente me considero más preparada y formada para afrontar futuros trabajos o proyectos que impliquen la programación.

Por lo general, estoy muy satisfecha con los conocimientos adquiridos y el resultado final que se ha obtenido, que ayudará, en mayor o menor medida, al LSI a avanzar en sus investigaciones, lo que le de mayor significado a todo el trabajo realizado.

Por último, señalar que el código de la herramienta es abierto y gratuito, y se puede obtener en el repositorio creado en *Bitbucket* [42]. Está a la disposición de la Universidad y sus investigadores para que pueda seguir desarrollándose y mejorando.

REFERENCIAS

[1] OBS Bussines School, "¿Qué es un diagrama de Gantt y para qué sirve?" https://goo.gl/x13edq [Último acceso: 10 Septiembre 2017].

[2] Wikipedia, "Vehículo autónomo", 20 septiembre 2017. https://goo.gl/4PPxGf [Último acceso: 21 septiembre 2017].

[3] Leire Pérez, 20minutos, "Los seis niveles de clasificación de los coches autónomos", 1 septiembre 2017. https://goo.gl/LxMuWc [Último acceso 21 septiembre 2017]

[4] 20minutos, "Audi A8, el primero en alcanzar una autonomía de nivel 3", 11 septiembre 2017. https://goo.gl/NuUdNS [Último acceso 22 septiembre 2017]

[5] Javier Sánchez, GQ, "Intel y BMW ya le han puesto fecha a la desaparición de los conductores: año 2021", 6 abril 2017. https://goo.gl/bv41DW [Último acceso 22 septiembre 2017]

[6] Juan Ranchal, Muycomputer, "Mercedes-Benz muestra su coche sin conductor F 015", 6 enero 2015. https://goo.gl/UjTjuL [Último acceso 22 septiembre 2017]

[7] M. Prieto, F. García, Expansión, "Google retira su coche autónomo", 16 junio 2017. https://goo.gl/hpKwNu [Último acceso 22 septiembre 2017]

[8] Javier Penalva, Xataka, "Sí, Apple por fin confirma que trabajan en el coche autónomo", 15 junio 2017. https://goo.gl/uavNAt [Último acceso 22 septiembre 2017]

[9] A. de la Escalera, J.M. Armingol, D. Martín Gómez, F. García, A. H. Al-Kaff, "Introducción a la visión por computador: desarrollo de aplicaciones con OpenCV" https://goo.gl/4DpuKa [Último acceso 13 septiembre 2017]

[10] Alberto Palomo, "¿Qué demonios es el Deep Learning y por qué debería aprenderlo en este artículo?", 2 septiembre 2017. https://goo.gl/XBpwq1 [Último acceso 18 septiembre 2017]

[11] J. Janai, F. Guney, A. Behla, A. Geiger, " Computer Vision for Autonomous Vehicles: Problems, Datasets and State-of-the-Art", arXiv:1704.05519, 18 Apr 2017.

[12] Girshick, R., Donahue, J., Darrell, T. and Malik, J. "ImageNet: Rich feature hierarchies for accurate object detection and semantic segmentation." arXiv:1311.2524. 2013.

[13] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J. and Zisserman, A. *"Pascal VOX Challenge: International Journal of Computer Vision, 88(2), 303-338, 2010"*

[14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Geiger, A., Lenz, P., & Urtasun, R. (2012, Junio). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*(pp. 3354-3361). IEEE.

[16] LabelImg, https://goo.gl/gE1CYc [Último acceso 17 septiembre 2017]

[17] LabelMe, https://goo.gl/Qdc6Cu [Último acceso 17 septiembre 2017]

[18] LEAR Image Annotation Tool
https://goo.gl/Wx8i82 [Último acceso 17 septiembre 2017]

[19] Ratsnake Image, https://goo.gl/GZSURV [Último acceso 17 septiembre 2017]

[20] RectLabel, https://goo.gl/McV6bm [Último acceso 18 septiembre 2017]

[21] VGG Image Annotator
https://goo.gl/R3MQLR [Último acceso 18 septiembre 2017]

[22] D. Martín, F. García, B. Musleh, D. Olmeda, G. Peláez, P. Marín, A. Ponz, C. Rodríguez, A. Al-Kaff, A. de la Escalera, J.M. Armingol, "iCab: Autonomous unmanned Ground Vehicle System"

[23] Wikipedia, "Sistema Operativo Robótico (ROS)", 22 noviembre 2016.
https://goo.gl/vZKA66 [Último acceso 18 septiembre 2017]

[24] D. Martín, F. García, B. Musleh, D. Olmeda, G. Peláez, P. Marín, A. Ponz, C. Rodríguez, A. Al-Kaff, A. de la Escalera, J.M. Armingol, "IVVI 2.0: An intelligent vehicle based on computational perception"

[25] Qt Creator, https://goo.gl/AUyjKA [Último acceso 9 septiembre 2017]

[26] Hipertextual, "13 razones para utilizar Qt", 14 marzo 2011
https://goo.gl/4zDAG3 [Último acceso 9 septiembre 2017]

[27] Qt Gui, Qt Creator, https://goo.gl/HzsSGX [Último acceso 9 septiembre 2017]

[28] OpenCV, https://goo.gl/6tFiP3 [Último acceso 10 septiembre 2017]

[29] TinyXML-2, https://goo.gl/Q6iuak [Último acceso 10 septiembre 2017]

[30] Point, Class Template Reference, OpenCV Classes
https://goo.gl/NuZeCA [Último acceso 6 septiembre 2017]

[31] mouseDoubleClickEvent, QWidget Class, Qt Documentation
https://goo.gl/dzLC2k [Último acceso 6 septiembre 2017]

[32] Drawing Functions, OpenCV Documentation
https://goo.gl/fKTU6S [Último acceso 6 septiembre 2017]

[33] boundingRect, Structural Analysis and Shape Descriptors, OpenCV Documentation.
https://goo.gl/J4RvbS [Último acceso 6 septiembre 2017]

[34] Rect, Class Template Reference, OpenCV Classes
https://goo.gl/UB3peB [Último acceso 6 septiembre 2017]

[35] Mat, Class Template Reference, OpenCV Classes
https://goo.gl/GeX7to [Último acceso 6 septiembre 2017]

[36] DGT, "Tráfico establece el marco para la realización de pruebas con vehículos de conducción automatizada en vías abiertas a la circulación", 16 noviembre 2015.
https://goo.gl/iZ2XtA [Último acceso: 20 septiembre 2017].

[37] Gizmodo, "En España ya es legal probar coches autónomos como los de Uber, y pronto será legal viajar en ellos", 12 diciembre 2016.
https://goo.gl/jZK4u4 [Último acceso: 20 septiembre 2017].

[38] Rosalía Sánchez, ABC, "Berlín aprueba el primer código ético del mundo para vehículos autónomos", 23 agosto 2017.
https://goo.gl/dJAvTc [Último acceso: 20 septiembre 2017].

[39] Intel, "Intelligent Driving: Experience a Ride with Intel Internet of Things", septiembre 2015.
https://goo.gl/XxaUed [Último acceso: 12 Septiembre 2017].

[40] D. D. Zetsche, *An interesting talk by the MD of Daimler Benz.* [Entrevista]. 10 mayo 2017.

[41] Sage, "La flexibilidad laboral", 21 marzo 2013.
https://goo.gl/X8SVXp [Último acceso: 14 septiembre 2017].

[42] Inés Carpio Coll, Bitbucket, "Repositorio de la herramienta de etiquetado denso desarrollada para el TFG", 7 julio 2017
https://goo.gl/DNB4Sp [Último acceso 25 septiembre 2017]